

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

MOBILNÍ APLIKACE PRO PREDIKCI SUBJEKTIVNÍ KVALITY ZÁŽITKU
S DATOVOU SLUŽBOU

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. RASTISLAV ČERVENÁK

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

MOBILNÍ APLIKACE PRO PREDIKCI SUBJEKTIVNÍ KVALITY ZÁŽITKU S DATOVOU SLUŽBOU

MOBILE APPLICATION FOR SUBJECTIVE QUALITY OF EXPERIENCE PREDICTION OF DATA
SERVICE

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. RASTISLAV ČERVENÁK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. JIŘÍ HOŠEK, Ph.D.

BRNO 2015



**VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky
a komunikačních technologií**

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Rastislav Červenák

ID: 125395

Ročník: 2

Akademický rok: 2014/2015

NÁZEV TÉMATU:

Mobilní aplikace pro predikci subjektivní kvality zážitku s datovou službou

POKYNY PRO VYPRACOVÁNÍ:

Diplomová práce se zabývá návrhem mobilní aplikace pro hodnocení kvality zážitku (Quality of Experience - QoE) v reálné mobilní síti. V teoretické části se bude nutné seznámit se způsoby hodnocení QoE a dále také specifiky QoE v prostředí mobilních sítí. Praktická část bude věnována návrhu aplikace pro měření QoE v mobilní síti. Jako vstupní parametry budou použity aktuální síťová propustnost a počáteční zpoždění daného spojení (odezva serveru). Jako výstup pak bude zobrazena hodnota MOS (Mean Opinion Score) vypočítaná na základě analytického modelu implementovaného do vytvořené aplikace. Součástí práce bude také návrh grafického prostředí dané aplikace a implementace funkce pro měření aktuální propustnosti a zpoždění. Následně pak bude implementován algoritmus pro výpočet MOS a ukládání získaných hodnot do databáze. Celková funkčnost celého systému bude ověřena sadou měření v reálném prostředí.

DOPORUČENÁ LITERATURA:

- [1] ITU-T Recommendation G.1030, "Estimating End-To-End Performance in IP Networks for Data Applications," Geneva, Switzerland, Tech. Rep., 2005.
- [2] FREEMAN, E.: Head First HTML5 Programming: Building Web Apps with JavaScript. Sebastopol: O'Reilly Media, 2011, ISBN: 978-1449390549.
- [3] NATILI, G.: PhoneGap 3, Beginner's Guide, 2nd Edition. Birmingham: Packt Publishing, 2013, ISBN: 978-1782160984.

Termín zadání: 9.2.2015

Termín odevzdání: 26.5.2015

Vedoucí práce: Ing. Jiří Hošek, Ph.D.

Konzultanti diplomové práce:

doc. Ing. Jiří Mišurec, CSc.

Předseda oborové rady

ABSTRAKT

Práca sa zaoberá problematikou užívateľskej spokojnosti s dátovou mobilnou službou. Hlavná časť je venovaná vývoju mobilnej aplikácie, ktorá zabezpečuje testovanie parametrov siete, ktoré vyhodnocuje a pomocou online databáze zdieľa s ostatnými užívateľmi. Rozobrané sú možnosti distribúcie aplikácie medzi užívateľov počas jej testovania a následne do App Store.

KĽÚČOVÉ SLOVÁ

QoE, QoS, mobilná dátová služba, mobilná aplikácia, iOS

ABSTRACT

This work deals with user satisfaction with mobile data service. The main part is devoted to the development of a mobile application that provides testing network parameters to evaluate and use online databases shared with other users. Discussed the possibility of distributing applications among users over its testing and then to the App Store.

KEYWORDS

QoE, QoS, mobile data network, mobile application, IOS

ČERVENÁK, Rastislav *Mobilní aplikace pro predikci subjektivní kvality zážitku s datovou službou*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2014. 58 s. Vedúci práce bol Ing. Jiří Hošek, Ph.D.

PREHLÁSENIE

Prehlasujem, že som svoju diplomovou prácu na tému „Mobilní aplikace pro predikci subjektivní kvality zážitku s datovou službou“ vypracoval samostatne pod vedením vedúceho diplomovej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej diplomovej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto diplomovej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/nebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona č. 121/2000 Sb., o právu autorskom, o právach súvisejúcich s právom autorským a o zmene niektorých zákonov (autorský zákon), vo znení neskorších predpisov, vrátane možných trestnoprávnych dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákoníka č. 40/2009 Sb.

Brno

.....

(podpis autora)

POĎAKOVANIE

Rád by som poďakoval vedúcemu semestrálnej práce pánu Ing. Jiří Hošek, Ph.D. za odborné vedenie, konzultácie, trpezlivosť a podnetné návrhy k práci a k vyvíjanej aplikácii.

Brno

.....

(podpis autora)

POĎAKOVANIE

Výzkum popsaný v tejto diplomovej práci bol realizovaný v laboratóriách podporených projektom SIX; registračné číslo CZ.1.05/2.1.00/03.0072, operačný program Výzkum a vývoj pro inovace.

Brno

.....
(podpis autora)

OBSAH

1	Úvod	11
2	Quality of Experience	13
2.1	Tvorba empirického modelu výpočtu QoE	14
2.2	Meranie Quality of Experience	15
2.2.1	Subjektívne zložky QoE	16
2.2.2	Objektívne zložky QoE	16
2.2.3	Vplyv užívateľského terminálu na QoE	16
2.2.4	Vplyv sieťového výkonu na QoE	16
3	Vytvorenie modelu pre výpočet MOS	18
4	Tvorba mobilnej aplikácie	19
4.1	Vývoj mobilnej aplikácie pre iOS	19
4.2	Implementácia algoritmov merania parametrov a získavanie dát o uživateli	19
4.2.1	Výpočet odozvy serveru na HTTP požiadavok	21
4.2.2	Výpočet prenosovej rýchlosti	23
4.2.3	Zobrazenie parametrov mobilnej dátovej siete	25
4.2.4	Zisťovanie polohy užívateľa	25
4.3	Tvorba dizajnového návrhu aplikácie	28
4.3.1	Auto Layout	28
4.3.2	Size Classes	28
4.3.3	Tvorba objektov a nastavenie grafického prostredia	29
4.3.4	Obrazovka Test	31
4.3.5	Obrazovka My History	31
4.3.6	Obrazovka Map	36
4.4	Práca s dátami z meraní	36
4.4.1	Online databáza	39
4.4.2	Core Data	42
4.5	Distribúcia aplikácie	43
4.5.1	Formy distribúcie iOS aplikácie medzi beta testerov	43
4.5.2	Distribúcia aplikácie do App Store	44
4.5.3	Monitorovanie distribuovanej aplikácie	44
5	Vyhodnotenie užívateľmi nameraných hodnôt	47
6	Závěr	55

Literatúra	56
7 Obsah přiloženého archívu	58

ZOZNAM OBRÁZKOV

2.1	Hodnotenie QoE na rôznych úrovniach	15
4.1	Graf prenosovej rýchlosti nameraný aplikáciou - porovnanie priemernej hodnoty a mediánu	21
4.2	Vývojový diagram začiatku merania	22
4.3	Meranie odozvy serveru na HTTP GET požiadavok	24
4.4	Vývojový diagram sťahovania a ukladania prenosovej rýchlosti	26
4.5	Porovnanie merania prenosovej rýchlosti s ďalšími aplikáciami	27
4.6	Funkcia constraints	29
4.7	Nastavovanie chovanie obrazovky pomocou Size Classes	30
4.8	Nastavenie chovania grafického objektu v PaintCode	31
4.9	Návrh ikon pre aplikáciu	32
4.10	Priebeh obrazovky Test	33
4.11	Obrazovka My History	34
4.12	Obrazovka My History Detail	35
4.13	Nastavovanie priorít pomocou Auto Layout	35
4.14	Obrazovka mapa	37
4.15	Obrazovka mapa error hláška	38
4.16	Filtre pre mapu	41
4.17	Návrh modelu pre Core Data	42
4.18	Ikony aplikácie pre všetky zariadenia	45
4.19	Ukážka služby Fabric.io	46
4.20	Zaznamenané udalosti pomocou služby mixpanel	46
5.1	Porovnanie parametru MOS pre mobilnú dátovú sieť a WiFi	48
5.2	Porovnanie prenosovej rýchlosti pre mobilnú dátovú sieť a WiFi	49
5.3	Porovnanie oneskorenia siete pre mobilnú dátovú sieť a WiFi	49
5.4	Porovnanie vplyvu jednotlivých rádiových technológií na MOS	50
5.5	Porovnanie českých operátorov v dosiahnutých výsledkoch z meraní od užívateľov	52
5.6	Porovnanie kvality internetu v meste a v jeho okolí	53
5.7	Porovnanie priemernej kvality mobilného internetu v rôznych krajinách	54

ZOZNAM TABULIEK

2.1	Porovnanie typov služieb a vplyvu QoS parametrov na ich kvalitu[2] .	14
4.1	Parametre odosielané na server	40
5.1	Zhrnutie dát o aplikácií	51

1 ÚVOD

Diplomová práca je venovaná problematike odhadu subjektívneho zážitku pri používaní dátovej služby pomocou mobilnej aplikácie. Tento zážitok priamo ovplyvňujú faktory, ktoré súvisia s kvalitou siete, ako je doba ktorú musí užívateľ čakať kým sa mu webová stránka začne načítavať, alebo rýchlosť sťahovania dát zo serveru služby. Užívateľa ale môžu ovplyvniť aj ďalšie faktory, a to je napríklad prostredie v ktorom sa nachádza, alebo kvalita zariadenia, ktoré používa. Práca sa bude zaoberať meraním parametrov siete, ktoré sú závislé na kvalite mobilnej dátovej siete. Tieto parametre sa spracujú a pomocou empiricky získaného modelu sa vypočíta kvalita internetového pripojenia. Vývoj mobilnej aplikácie pre platformu Apple iOS, ktorá všetky dáta získa a spracuje je hlavným cieľom celej práce, na ktorej výstupe je funkčná aplikácia.

V prvej časti sa práca venuje problematike hodnotenia kvality zážitku (Quality of Experience) ako takej a pre špecifikácie mobilnej siete a mobilných terminálov. Zaoberá sa možnosťami a spôsobmi merania QoE.

Potom sa práca venuje spôsobu vývoja mobilnej aplikácie, a to obecné a konkrétne pre platformu Apple iOS. Rozobrané sú možnosti prístupu návrhu aplikácie a prostriedky platforme iOS ktoré vývojárom ponúka. Pri vývoji mobilnej aplikácie sa musí dbať na jej dizajn a užívateľský predvídateľné prostredie pre konkrétnu platformu. Aplikácia by sa mala prispôbovať vždy konkrétnemu zariadeniu, teda jeho rozlíšeniu a jeho výpočtovému výkonu. Interpretácia dát by mala byť zrozumiteľná. Všetko toto je riešenie v časti venovanej dizajnu.

Samotná aplikácia využíva základne, ale aj externé knižnice pre beh aplikácie. Rieši komunikáciu so serverami, kontroluje aktuálny stav a podmienky aplikácie, ako je pripojenie k sieti, polohové služby, užívateľské nastavenia a ukladá namerané informácie do online databáze a tiež do zariadenia užívateľa. Počas merania sa algoritmy na meranie rýchlosti prispôbujú aktuálnej rýchlosti internetu. Popis ako to robí je v kapitole o vývoji aplikácie.

Keďže aplikácia neslúži iba k zobrazeniu aktuálneho výsledku merania, ale aj zdieľaných týchto dát, vytvoril som spojenie s online databázou. Aplikácia s ňou komunikuje pre účely ukladania dát a ich získavania. Tomu, ako s ňou prebieha komunikácia a ako je nakonfigurovaný server je venovaná časť o databáze. V nej je zároveň rozobraný spôsob ukladania užívateľských dát do pamäti zariadenia.

Aby sa overila funkčnosť aplikácie spravilo sa jej beta testovanie na rôznych užívateľoch. To aké sú možnosti distribúcie užívateľom vývojárskej verzie a následne distribúcie aplikácie do App Store je ukázané v poslednej časti. Mimo iné zobrazuje aj možnosti sledovania spätnej odozvy od užívateľov a automatické zaznamenávanie chovania užívateľov v aplikácií a reportovanie pádov aplikácie.

Jednotlivé algoritmy, ktoré meranie a reprezentáciu výsledkov vykonávajú sú popísané pomocou vývojových diagramov.

2 QUALITY OF EXPERIENCE

Quality of Experience (QoE), teda kvalita zážitku zo služby, môžeme definovať to, ako užívateľ vníma použiteľnosť alebo stupeň spokojnosti so službou. ITU-T[1] definuje QoE ako celkovú prijateľnosť aplikácie alebo služby tak, ako je subjektívne vnímaná koncovým užívateľom. Sú v ňom zahrnuté účinky celého systému od začiatku až do konca (klient, terminál, sieť, služby infraštruktúry, atď.). Celková prijateľnosť môže byť ovplyvnená užívateľskými očakávaniami a kontextom. Iná definícia hovorí o stupni radosti alebo nespokojnosti užívateľa aplikácie alebo služby. Vyplýva to z plnenia jeho očakávaní so službou vplyvnom použitých nástrojov s ohľadom na osobnosť užívateľa a jeho aktuálnom stave.

Kvalitu služby je možné určovať aj pomocou Quality of Service (QoS). Tá vyjadruje schopnosť siete poskytovať služby so zaistením servisnej úrovne. QoS je meraná z hľadiska priepustnosti, straty paketov a oneskorenia. Je to charakteristika siete so zameraním na kvalitu služieb, zatiaľ čo QoE je charakteristika kvality služieb so zameraním na subjektívny dojem užívateľa.

QoE teda závisí na ľudských aj technických faktoroch. Užívateľ má iné očakávania pre rôzne druhy služieb. To ako ho budú jednotlivé faktory ovplyvňovať jeho spokojnosť závisí mnohokrát aj na jeho znalostiach a skúsenostiach so službou. Preto niektoré faktory nemusia mať na jeho spokojnosť vôbec vplyv a naopak niektoré ju môžu ovplyvňovať viac ako iných. Tieto faktory sa dajú rozdeliť na:

- **Aplikačné faktory:** napríklad VoIP, video streaming, prístup na web, ale tiež charakteristiky protokolov UDP, TCP, HTTP, ktoré ovplyvňujú ako služba funguje a čo od nej môžeme očakávať. Napríklad pri spojení TCP očakávame bezchybný prenos, pri HTTP korektné zobrazenie informácie a sťahovanie súborov z jedného zdroja. Pri sťahovaní súborov cez BitTorrent protokol sa súbor môže sťahovať sekvenčne, a tak výsledný súbor môže byť dostupný až po stiahnutí všetkých dát.

Pre každý druh služieb sú iné očakávania od služby. V tab 2.1 sú uvedené príklady služieb, ktorých výslednu kvalitu hodnotí užívateľ podľa faktorov, ktoré ovplyvňuje napríklad aj oneskorenie alebo stratovosť. To sú parametre QoS, čo ukazuje priamu súvislosť medzi QoE a QoS. Vyplýva z nej, že pri hlasových správach akceptujeme chyby pri prenose, ale vyžadujeme malé oneskorenie. Naproti tomu služby na pozadí, ako je kontrola mailov môže mať veľké oneskorenie, ale priate dáta musia byť bez chýb. Ak to tak nie je, užívateľ môže so službou nespokojný.

- **Obsah:** napríklad video alebo video konferencia a ich kvalita.
- **Zariadenie užívateľa** a jeho vlastnosti: rozlíšenie obrazovky, prostredie, výdrž batérie a výkonnosť.

- **Sieťové podmienky:** stratovosť paketov, oneskorenie, prenosová rýchlosť
- **Charakteristika služieb:** spoľahlivosť, dostupnosť, pokrytie, cena a pod.
- **Kontext:** ktorý môže zahŕňať typ prístupu (bezdrôtový/káblový), pohyblivosť (mobilný/stacionárny), lokalizácia a pod.

Tab. 2.1: Porovnanie typov služieb a vplyvu QoS parametrov na ich kvalitu[2]

Chybe tolerantné	Konverzia zvuku a video	Hlasové/Video správy	Streaming audio a video	Fax
Chybe netolerantné	Ovládanie príkazmi (Telnet, interaktívne hry)	Transakcie (elektronické bankovníctvo, webové prehliadanie, prístup k mailu)	Textová komunikácia v reálnom čase, sťahovanie súborov (FTP, iMessage)	Služby na pozadí (kontrola aktualizácií)
	Interaktívne (oneskorenie << 1 s)	Reagujúce (oneskorenie ~ 2 s)	Aktuálne (oneskorenie ~ 10 s)	Nie kritické oneskorenie >> 10 s

2.1 Tvorba empirického modelu výpočtu QoE

Na to, aby sme vedeli určiť očakávanú spokojnosť užívateľa na základe merateľných parametrov je potrebné vytvoriť model pomocou ktorého sa dá predpokladať výsledok. Samotné QoE pri čiste technickom meraní nemusí byť presné, preto je lepšou možnosťou vytvoriť empirický model a najlepšie s čo najväčším množstvom objektov. Tento postup je ale obmedzený z organizačných a finančných dôvodov.

Pri vytváraní modelu môžeme využiť zmerané subjektívne skúsenosti a na základe nich rozvíjať predikčný model QoE. Tieto modely sú potom ďalej používané ako súčasť vnímania kvality na základe podkladových podmienok systému. Vieme potom napríklad odhadnúť aký bude mať vplyv BER (Bit Error Rate - pomer počtu chybných prijatých bitov k celkovému počtu prijatých bitov) na to, ako bude užívateľ vnímať službu. Vždy ide ale o individuálne vnímanie, preto tieto modely nemôžu byť nikdy úplne presné[4].

Model sa vytvára pomocou obecných faktorov, ktoré QoE ovplyvňujú. Sú to kontext, samotný užívateľ, systém a obsah.

Kontext

Kontext je prezentovaný aktuálnym reálnym scenárom v ktorom sú služby používané. Toto prostredie má samo o sebe značný vplyv, a to priamy alebo nepriamy, na QoE pretože určuje zobrazovaciu vzdialenosť, sociálne a kultúrne pozadie a iné.

Užívateľ

QoE popisuje ako užívateľ vníma aplikáciu alebo službu relatívne k jeho očakávaniám. Je významne ovplyvnení jeho vnútornými stavmi a predispozíciami. Je to práve jeho očakávanie, ktoré výrazne definuje výsledný dojem z kvality služby.

Systém

Vlastnosti technickej implementácie priamo ovplyvňuje QoE. Pre poskytovateľa služby tieto vlastnosti odráža ukazovateľ výkonnosti sieťových parametrov - Network Key Performance Indicators (KPI).

Kvalitu prijímaného obsahu charakterizuje užívateľské zariadenie (napríklad rozlíšenie displeja, jeho veľkosť a kvalita) a možnosti pripojenia do siete.

Obsah

Obsah a jeho spracovanie je výrazný faktor ovplyvňujúci celkový QoE. Obsah sa môže automaticky prispôbovať aktuálnym podmienkam. Napríklad u videa sa môže dynamicky meniť jeho kvalita v závislosti na kvalite pripojenia a tým predchádzať čakaniu na načítanie časti videa do pamäti.

Na obrázku č. 2.1 sú zobrazené parametre na úrovni siete, zariadenia a užívateľa, ktoré ovplyvňujú výsledný QoE.



Obr. 2.1: Hodnotenie QoE na rôznych úrovniach

2.2 Meranie Quality of Experience

QoE môže byť odhadnuté subjektívnym alebo objektívnym prístupom. Subjektívny odhad QoE vyžaduje zapojenie užívateľov a kvantifikuje QoE v podmienkach Mean

Opinion Score (MOS), kde je kvalita hodnotenia pomocou 5-bodovej stupnice: 5 = vynikajúci, 4 = výborný, 3 = dobrý, 2 = slabý, 1 = zlý. Na druhej strane objektívne meranie odhaduje QoE pomocou empirického modelu bez nutnosti zapojenia užívateľov. Empirický model závisí na aplikácii, kontextu a pod. a je funkciou QoS na úrovni siete, ktorá sa zvyčajne získava na základe merania.[6]

Skúsenosti užívateľov sú do značnej miery ovplyvnené parametrami ich terminálov. Preto sa pri analýze QoE môžu zbierať aj informácie o týchto termináloch.

2.2.1 Subjektívne zložky QoE

Zaraďujeme sem zložky, ktoré sa nedajú presne odmerať a výsledok závisí od každého jednotlivca, aktuálnej situácie a subjektívnej skúsenosti. Výsledkom toho je, že pri službe s identickými parametrami budú dvaja rôzni užívatelia vnímať kvalitu tejto služby rôzne, prípadne ju môže vnímať jeden užívateľ inak pri rôznych situáciach. Práve preto je dobré pri meraní QoE získať hodnoty od čo najväčšieho vzorku užívateľov. [3]

2.2.2 Objektívne zložky QoE

Objektívne zložky QoE reflektujú kvalitu služby vnímanú užívateľom, ktorý ju však môže hodnotiť cez objektívne metódy ako sú MDI (Media Delivery Index = poskytuje presné meranie kolísania a oneskorenia na úrovni siete), SSIM (Structural Similarity Index = vyjadruje podobnosť dvoch obrazov, metóda porovnávania), PEVQ (Perceptual Evaluation of Video Quality = perцепčné hodnotenie kvality videa), rýchlosť načítania webovej stránky a ďalšie. 2.1

2.2.3 Vplyv užívateľského terminálu na QoE

Odlišný výkon užívateľského terminálu, napríklad mobilného telefónu, môže mať vplyv na to, aký bude užívateľský zážitok z hodnotenia služby užívateľom.

Ovplyvňujúce faktory sú možnosti zariadenia k pripojeniu do siete (2G, 3G, LTE), rýchlosť spracovania prijatých/odoslaných dát, a tiež jeho výkon. Ten závisí na použití hardwaru, ako je procesor, grafický adaptér, užívateľský hardwarový interface, ale aj na softwarom spracovaní.

2.2.4 Vplyv sieťového výkonu na QoE

Výkon a vyťaženosť siete má priamy vplyv na to, ako užívateľ vníma službu. Pri meraní QoE sa merajú QoS sieťovej služby a na jej základe sa dá predpokladať výsledok a vytvárať model predikcie QoE.

Najčastejšie sa uvažuje šírka pásma, oneskorenie a bitová chybovosť. Vždy záleží na konkrétnej službe ktorú užívateľ chce využívať, ktorý z týchto parametrov bude pre neho najdôležitejší. Pri telefonovaní cez internet (VoIP) je dôležitejšie oneskorenie ako prenosová šírka pásma, naopak pri službe video na vyžiadanie je potrebná dostatočná rýchlosť prenosu dát a mierne oneskorenie má na zážitok zo služby u užívateľa menší vplyv.

3 VYTVORENIE MODELU PRE VÝPOČET MOS

Pri vytváraní modelu pre výpočet parametru MOS je nutné brať vplyv subjektívneho vnímania jednotlivcov, ktoré je ovplyvnené mnohými faktormi. Metódy pre testovanie subjektívnej kvality popisuje dokument ITU-T Rec. P.800.[7] Jedna z metód je metóda ACR (Absolute Category Rating), kde sa prezentuje testovaná vzorka každému jednotlivcovi. Tí potom podľa stupnice kvality hodnotia ich dojem z každého vzorku. Tieto vzorky môžu byť hodnotené z rôznych hľadísk. Stupnica hodnotenia kvality je zvyčajne zvolená od 1 do 5.

V práci Predicting user QoE satisfaction in current mobile networks [8] je popis tvorby empiricky získaného vzorca, ktorý predikuje užívateľský zážitok z dátovej siete na základe prenosovej rýchlosti a oneskorenia siete. Tento výskum bol robený na vzorke ľudí rôzne diverzifikovaných podľa pohlavia, veku, vzdelania a dosiahnutého vzdelania. Parametre QoS, ktoré určovali kvalitu siete boli prenosová rýchlosť a doba načítavania.

Testované objekty boli vždy pri zmene parametrov siete požiadané o hodnotenie ich dojmu zo služby pomocou stupnice MOS, teda od 1 pre zlý a 5 pre vynikajúci dojem. Testovaný obsah odrážal skutočné webové služby a sťahovanie súborov. Pri testovaní sa menili scenáre kombináciou rôznych dátových tokov a počiatočného oneskorenia siete.

Dáta z týchto meraní boli následne spracované a bol vytvorený model pomocou regresnej analýzy. To je metóda matematickej štatistiky, ktorá odhaduje pravdepodobnosť nejakého javu na základe určitých známych skutočností. Klasický lineárny model nebol vhodný, pretože v tomto prípade neplatí lineárny vzťah medzi parametrami siete a výsledným hodnotením užívateľov. Použitá teda bola logaritmická regresia. Model charakterizuje odhadované hodnotenie MOS podľa vstupných parametrov, a to prenosovej rýchlosti siete a doby načítania stránky. Vzorec je v práci využitý pre výpočet odhadovaného užívateľského zážitku mobilnou aplikáciou a je nasledovný:

$$MOS = \frac{(b - a)}{1 + e^{-c_0 + c_1 * D * c_2 * \log(BR)}} + a$$

$$c_0 = -0.6014, c_1 = -0.0963, c_2 = -0.2748$$

D [sekunda] - Oneskorenie siete

BR [kbit/s] - Rýchlosť sťahovania z internetu

4 TVORBA MOBILNEJ APLIKÁCIE

Vyvíjaná aplikácia bude slúžiť na meranie parametrov siete ako je doba odpovedi zo serveru na HTTP požiadavok a maximálna prenosová rýchlosť sťahovania súboru z internetu, ktorej výsledkom je hodnota QoE vyjadrujúca kvalitu zážitku z dátovej služby. Aplikácia bude určená pre mobilné zariadenia na platforme Apple iOS. Jej výstupom po skončení merania budú hodnoty nameraných hodnôt ktoré sa zobrazia užívateľovi s možnosťou porovnať ich s hodnotami iných užívateľov na mape.

4.1 Vývoj mobilnej aplikácie pre iOS

Pre užívateľa by mala byť aplikácia čo najjednoduchšia prehľadná. Je dobré aby využívala princípy užívateľského prostredia, na ktoré sú užívatelia pre danú platformu zvyknutí. Mali by byť vždy informovaní o tom, čo aplikácia práva vykonáva. V niektorých prípadoch je dobré spúšťať procesy vo viacerých vláknach. Tým sa zamedzí tzv. zamrznutiu aplikácie, kedy ostatné procesy musia čakať na dokončenie iného procesu. Typickým príkladom môže byť komunikácia zo serverom a čakanie na jeho odpoveď na dotaz.

Pri vývoji iOS aplikácie sa musí počítať s obmedzeniami, ktoré táto platforma dáva. Každá aplikácia pracuje vo vlastnom prostredí v pamäti a až na výnimky nemá povolené pristupovať k dátam ostatných aplikácií. Pri prístupe k hardwarovým funkciám sa môžu používať iba verejné triedy a frameworky. Jazykmi tejto platformy je objektovo orientovaný programovací jazyk Objective-C 2.0 a/alebo Swift. Pre iOS vznikla nadstavba Cocoa a Cocoa Touch framework. Je to User Interface framework pre vývoj software na operačný systém iOS.

Foundation framework definuje základnú vrstvu Objective-C triedy. Umožňuje pracovať s objektami pre ukladanie dát, ako sú polia (NSArray), slovníky (NSDictionary), textové reťazce (NSString), časové formáty (NSDate) a podobne. Používa sa aj pri prístupe k hardwarovým funkciám zariadení pomocou ich API.

Natívne aplikácie pre iOS sa vyvíjajú vo vývojom prostredí Xcode. Obsahuje nástroje pre tvorbu a editáciu kódu, debugovanie, tvorbu dizajnu aplikácie, ladenie, simulačné nástroje pre testovanie aplikácií a distribúciu aplikácie do App Store.

4.2 Implementácia algoritmov merania parametrov a získavanie dát o užívateľovi

Pri implementácii jednotlivých algoritmov pre meranie prenosovej rýchlosti internetu a jeho oneskorenia som vytvoril objekty, ktoré zabezpečujú po ich zavolaní

vykonanie testu, informovanie objektu, ktorý si test vyžiadal, o aktuálnom stave merania a spracovanie nameraných hodnôt do požadovaného formátu. Niektoré objekty využívajú otvorené knižnice pre prístup k informáciám ako je meno operátora vlastníaci SIM kartu užívateľa. Počas vývoja som využíval aj knižnicu na zisťovanie výkonu signálu k pripojenej BTS stanici. Táto knižnica ale nie je verejná a pri pokuse o nahratie aplikácie do App Storu už na beta testovanie bola aplikácia automaticky zamietnutá. Preto som túto funkciu z aplikácie odstránil úplne.

Meranie sieťovej priepustnosti

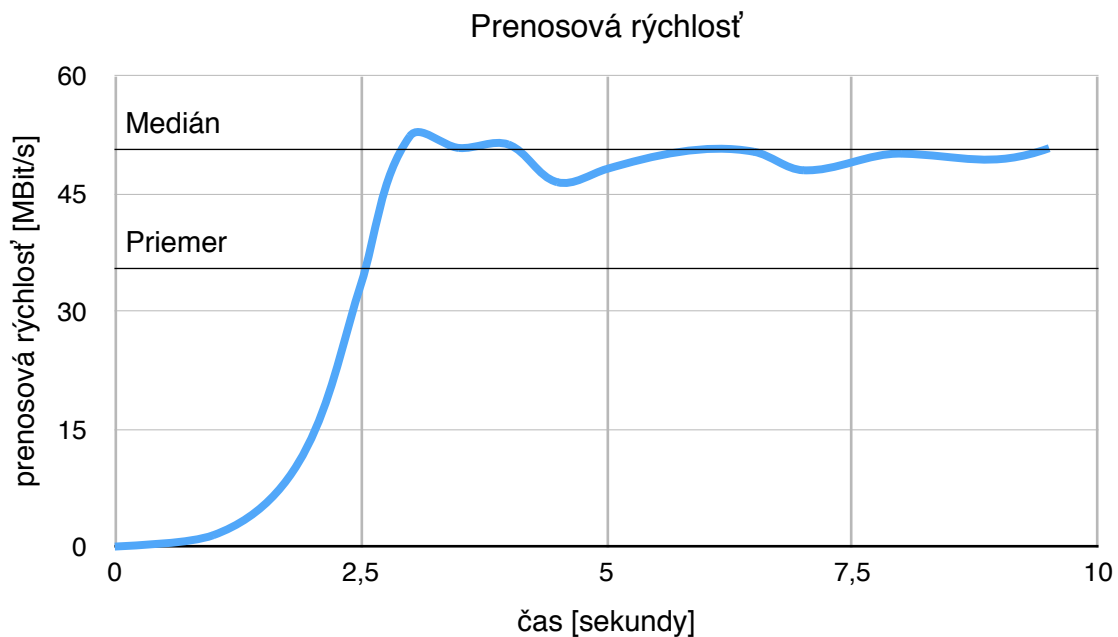
Pre implementáciu funkcie pre meranie sieťovej priepustnosti som využil triedy, ktoré ponúka priamo Foundation Framework.

Aplikácia si pri spustení merania overuje či je pripojenie k internetu aktívne. Zisťuje sa to pomocou natívnych metód iOS. Na ošetrovanie odozvy od serveru, ktorý môže byť nedostupný, je daný limit čakania odpovede na 60 sekúnd. Ak odozva zo serveru nepríde užívateľ je o tejto skutočnosti informovaný pomocou objektu `AlertView`, teda vsakovacie okno, a meranie je ukončené. Kontrola dostupnosti pripojenia prebieha počas celej doby merania. Tým je zároveň ošetrovaný prípad, ak by z nejakého dôvodu došlo k odpojeniu od internetu. Ak je pripojenie dostupné spustí sa po užívateľskom požiadavku na vykonanie testu meranie dĺžky doby odpovedi zo serveru na načítanie internetovej stránky a po jeho skončení tohoto testu sa spustí test rýchlosti sťahovania súboru.

Pri práci so sieťovými udalosťami využívam delegované metódy `NSURLConnectionDataDelegate` a `NSURLConnectionDelegate`, ktoré sú volané pri preddefinovaných udalostiach, napríklad: Prijatie odpovedi zo servera (`didReceiveResponse`), prijatie dát (`didReceiveData`) alebo problém pri spojení so serverom (`didFailWithError`).

Výsledné hodnoty oneskorenia siete a rýchlosti sťahovania dát sú prezentované ako medián všetkých hodnôt zaznamenaných počas merania pre jednotlivé testy. Medián som zvolil ako lepšie vyjadrenie výslednej hodnoty pred priemerom. Štatisticky vyjadruje mieru centrálnnej tendencie, a preto je pri praktickom meraní z obmedzenou dĺžkou merania presnejší. Ako príklad uvediem jedno reálne meranie spravené aplikáciou zobrazené na obrázku č. 4.1. Medián tejto hodnôt ktoré sú zobrazené na grafe je 49,41, priemerná hodnota je potom 39,44. Nižšia priemerná hodnota je spôsobená v tomto prípade hlavne nižšími hodnotami v prvej časti grafu. Táto hodnota ale nezodpovedá reálnej prenosovej rýchlosti, ktorú užívateľ môže využívať.

Algoritmus merania je zobrazený na obrázku č. 4.2

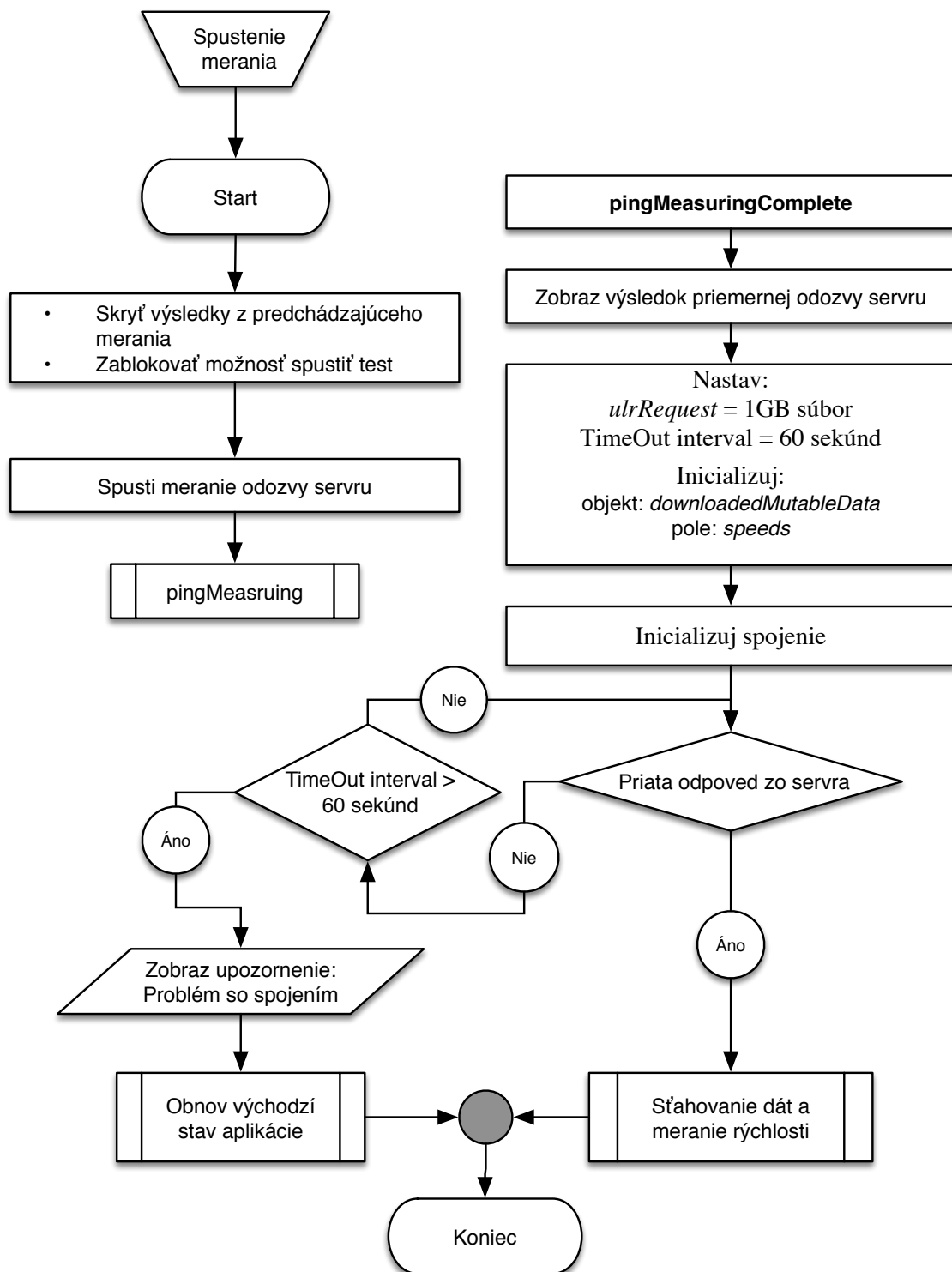


Obr. 4.1: Graf prenosovej rýchlosti nameraný aplikáciou - porovnanie priemernej hodnoty a mediánu

4.2.1 Výpočet odozvy serveru na HTTP požiadavok

Pomocou vytvoreného objektu z rodiča `NSURLConnection` sa vytvorí spojenie so serverom. V tom istom momente je spustený časovač vo funkcii `stopiek`, ktorý skončí, po zavolaní delegovanej metódy `NSURLConnection didReceiveResponse`. Tá je zavolaná, ak je po požiadavku na vytvorenie spojenia vrátená odpoveď zo servera. Aplikácia skontroluje, či je odpoveď kladná, teda HTTP so status kódom 200. Pre užívateľa táto hodnota znamená, že po tejto dobe by sa mu začala načítavať internetová stránka v jeho prehliadači. V prípade, že je odpoveď kladná, uloží sa nameraný čas, ukončí sa vytvorené spojenie a podľa potreby sa opakuje meranie znova. To, či sa má opakovať závisí od zadaného počtu testovania tohoto parametru. Zvolil som hodnotu 5, ako dostatočný počet na zistenie výslednej hodnoty v kompromise s tým, aby meranie nebolo príliš dlhé na pomalom pripojení. Jedno meranie totiž môže trvať aj niekoľko sekúnd. V druhom prípade, ak príde negatívna odpoveď zo servera, ukončuje sa celé meranie a užívateľ je o tom informovaný.

Na konci celého cyklu meraní oneskorenia sa všetky uložené hodnoty prevedú na medián a táto hodnota sa odošle objektu, ktorý si meranie vyžiadal. Ten sa postará o to, aby sa zobrazila výsledná hodnota užívateľovi a po ukončení celého merania odoslal hodnoty na ďalšie spracovanie. Objekt odosiela každú nameranú hodnotu aj po ukončení cyklu merania, aby bol užívateľ informovaný o hodnotách už počas



Obr. 4.2: Vývojový diagram začiatku merania

merania. Spolu s hodnotou sa odosiela aj poradová hodnota prebiehajúceho cyklu, a tá je vyjadrovaná pomocou indikátoru priebehu na obrazovke Test.

Server s ktorým aplikácia naväzuje spojenie je www.bbc.com, ktorého internetová stránka obsahuje obsah prezentujúci štandardnú internetovú stránku. Je pri nej zároveň predpoklad vysokej dostupnosti a distribuovanej infraštruktúry pre celý svet, a teda výsledné hodnoty by nemali byť ovplyvnené výraznými vzdialenosťami od servera pre rôzne miesta na svete.

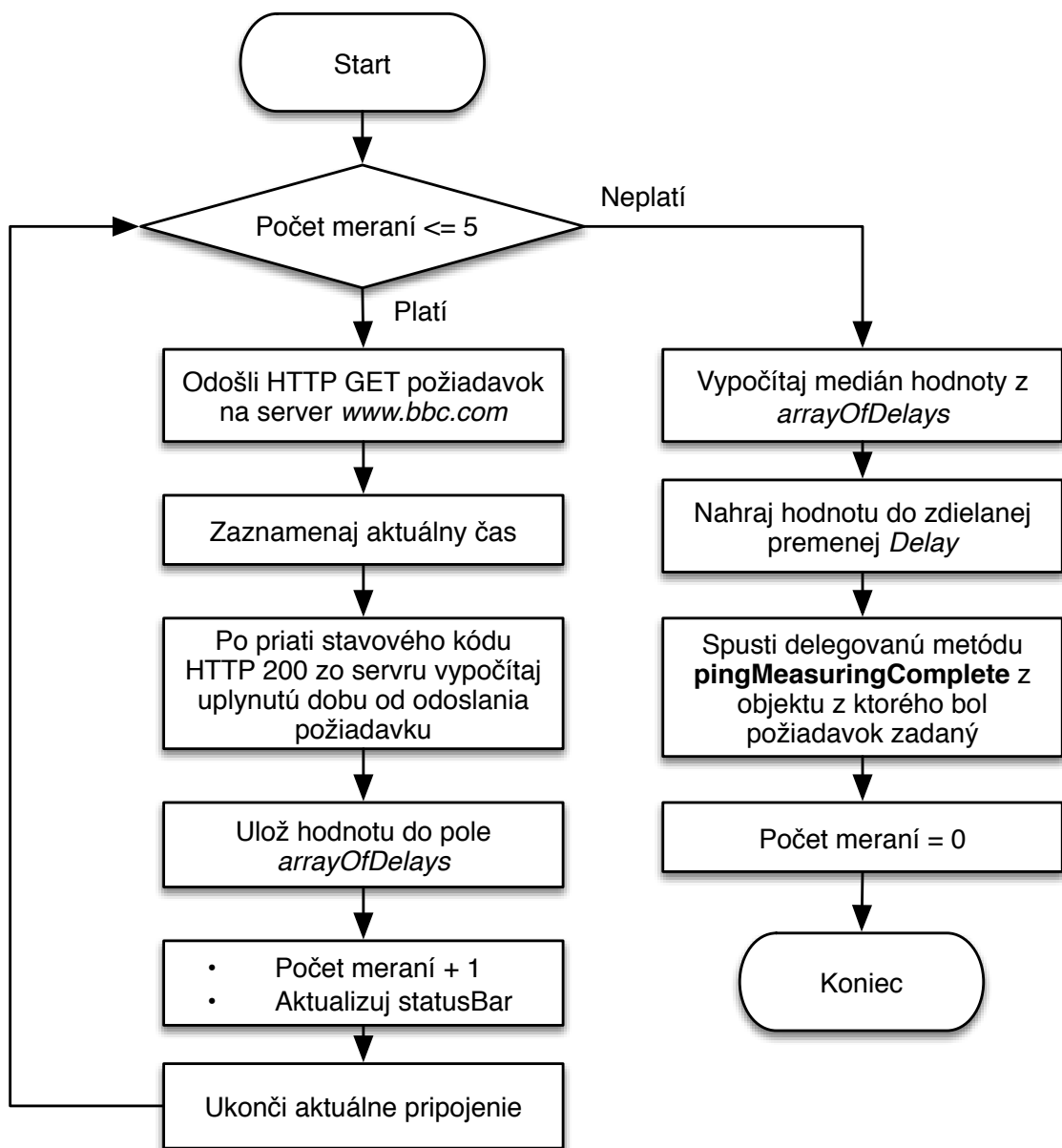
Algoritmus merania je zobrazený na obr. č. 4.3

4.2.2 Výpočet prenosovej rýchlosti

Meranie prenosovej rýchlosti začína podobne ako meranie doby oneskorenia. Vytvorí sa pomocou dediča objektu `NSURLConnection` spojenie s dopredu známou webovou adresou na server <http://speedtest.tweak.nl/1000mb.bin>. Veľkosť sťahovaného súboru som zvolil 1GB a dobu sťahovania súboru zo serveru, počas ktorého sa meria prenosová rýchlosť internetu, 10 sekúnd. Aby táto doba bola dostatočná na stiahnutie celého súboru, musela by byť rýchlosť sťahovania takmer 1Gbit/s. Preto sú algoritmy navrhnuté aplikácie vytvorené tak, že berú túto veľkosť súboru ako nekonečnú. a čaká sa na odpoveď zo servera, ktorá je zaznamenaná delegovanou metódou `didReceiveResponse`. V tejto chvíli je spustený časovač, ktorý je nastavený na 10 sekúnd. Táto doba bola opäť zvolaná ako kompromis pre dĺžku merania a jeho presnosť. Služi aj ako obmedzenie množstva dát, ktoré užívateľ stiahne počas merania.

Prenosová rýchlosť sa vzorkuje počas celej doby merania. Počíta sa ako množstvo prijatých dát od posledného vzorku $n-1$ po aktuálny n za dobu frekvencie vzorkovania. Tá je nastavená na 0,5 sekúnd. Týmto spôsobom je možné získať presnejšiu hodnotu rýchlosti ako pri výpočte rýchlosti za celú dobu sťahovania. Výsledná hodnota prenosovej rýchlosti sa určí pomocou mediánu všetkých nameraných hodnôt.

Pri pomalom pripojení môže vzniknúť problém dlhšej doby čakania na ďalšie dáta ako je frekvencia vzorkovania, a tým vznikajúcich nulových hodnôt meranej veličiny. Ak je takýchto hodnôt viac ako polovica všetkých vzorkov, medián je 0. Tieto vzorky nemôžu byť odstránené, pretože by sa zvýšila hodnota rýchlosti sťahovania, keďže hodnoty jednotlivých vzorkov sú považované za prenosovú rýchlosť za dobu 0,5 sekúnd. Na odstránenie, respektívne doplnenie týchto nulových vzorkov využívam interpoláciu posledného nenulového vzorku $n-1$ a nasledujúceho $n+1$. Tento postup je možný až po ukončení merania, keď už sú k dispozícii všetky hodnoty. Aby sa ale užívateľovi zobrazovali aspoň približné hodnoty aj počas merania využívam extrapoláciu dvoch posledných vzorkov $n-2$ a $n-1$. Interpolácia aj extrapolácia je využívaná iba ak sa v meraní vyskytne nulová hodnota prenosovej rýchlosti.



Obr. 4.3: Meranie odozvy serveru na HTTP GET požiadavok

Rýchlosť sa počíta ako:

$$\text{rýchlosť stahovania} = \frac{\text{stiahnuté dáta}[n] - \text{stiahnuté dáta}[n - 1]}{\text{vzorkovací čas}} \quad (4.1)$$

Algoritmus je popísaný v diagrame na obrázku č. 4.4.

Aby som overil, že aplikácia prenosovú rýchlosť správne, spravil som testy zároveň pomocou ďalších dvoch mobilných aplikácií OOKLA Speedtest a DSL.cz. Všetky tri aplikácie boli nainštalované na rovnakom zariadení, a testy boli vykonané s rovnakými podmienkami, teda na rovnakom mieste, rovnakou pripojenou mobilnou technológiou a merania boli pustené zároveň v slede za sebou. Z grafu č. 4.5 je vidno, že moja aplikácia meria veľmi podobne ako iné. Rozdiely môžu byť spôsobené neustálym kolísaním výkonu siete, aktuálnym vyťažením serverov, z ktorých sa dáta pre potreby merania sťahujú a tiež odlišnosťami v metodike pre konečný výpočet hodnoty.

4.2.3 Zobrazenie parametrov mobilnej dátovej siete

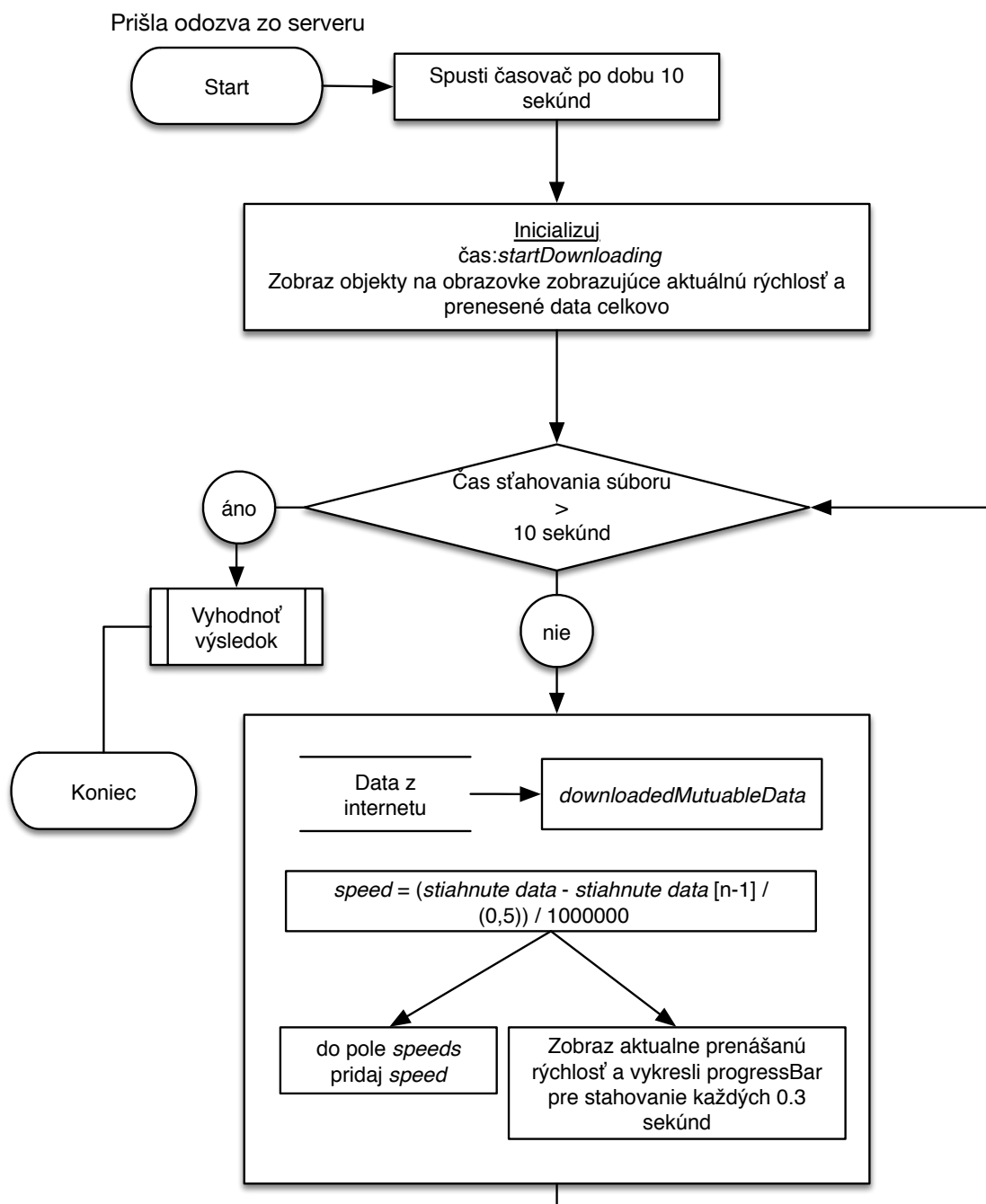
Zisťovanie parametrov mobilnej siete je riešené pomocou objektov tried z CoreTelephony. Pomocou nich som vytvoril privátne metódy, ktoré vracajú parametre, ako sú názov mobilnej siete do ktorej je telefón prihlásený, mobilnú dátovú technológiu do ktorej je zariadenie prihlásené, krajinu operátora vlastníaci domácu mobilnú sieť. Všetky tieto hodnoty sú ukladané do databáze spolu s nameranými hodnotami.

4.2.4 Zisťovanie polohy užívateľa

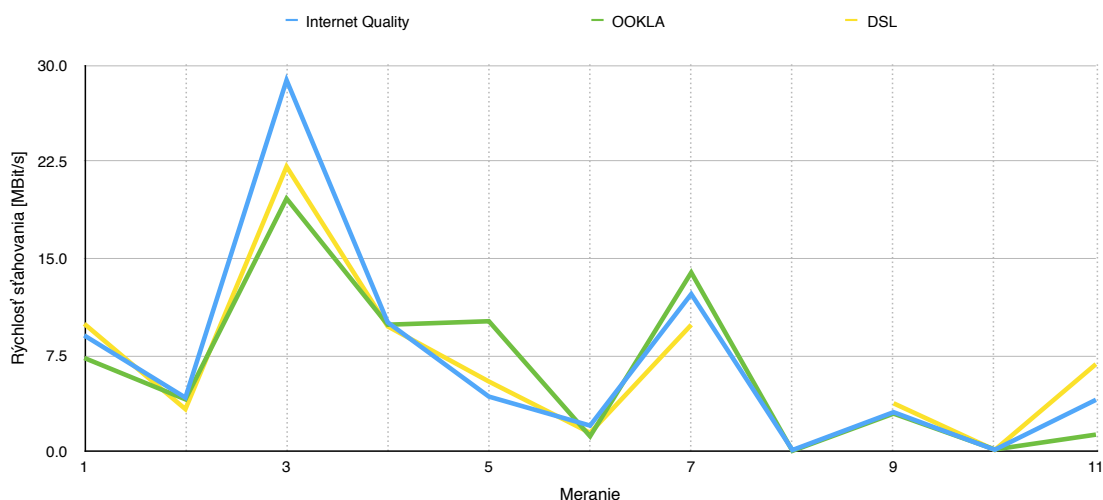
Pre aplikáciu vrátane jej online databázovej služby je dôležité, aby bola správne lokalizovaná krajina v ktorej sa užívateľ nachádza. Nie je to ale možné robiť iba pomocou metód z CoreTelephony, ktorá určuje krajinu a operátora podľa vlozenej SIM karty v zariadení, a to z dôvodov:

- Odstrániť roamingových užívateľov. Parametre z triedy CoreTelephony zisťuje názov siete a krajinu podľa operátora, ktorému patrí SIM karta. Preto aj keď bude užívateľ prihlásený do siete zahraničného operátora bude táto trieda vracaať operátora domáceho. Tým by sa dostávali do databáze hodnoty pre operátorov mimo územia v ktorom vysielajú.
- Užívateľ nemusí mať v zariadení žiadnu SIM kartu, ale pre účely obrazovky Map je nutné zistiť v akej krajine sa nachádza, aby sa dali načítať informácie do filtru pre túto krajinu.

V prípade ak má užívateľ povolené polohové služby, Apple umožňuje pomocou reverznej geolokačnej služby zistiť vlastnosti polohy užívateľa na základe jeho GPS



Obr. 4.4: Vývojový diagram sťahovania a ukladania prenosovej rýchlosti



Obr. 4.5: Porovnanie merania prenosovej rýchlosti s ďalšími aplikáciami

polohy. Týmto spôsobom je zistený ISO kód krajiny. Táto služba je online a počet požiadavkov na ňu je obmedzený. Aplikácia teda zisťuje tento parameter iba v prípade potreby a následne je tento údaj a uloží ho do globálnej premennej, takže ho už počas behu aplikácie nie je nutné zisťovať.

Ak užívateľ nepovolil polohové služby pre aplikáciu, alebo by sa nepodarilo zistiť kód krajiny predchádzajúcim spôsobom, vyžiada sa táto informácia podľa IP adresy. Pre tieto potreby som vytvoril skript na serveru, ktorý preberie parameter užívateľskej IP adresy a pomocou služby tretej strany získa túto informáciu. Informácia je získavaná vo forme JSON formátu. Skript na serveri tento JSON spracuje a získa z neho iba požadovanú informáciu, ktorú odošle ako odpoveď na požiadavok aplikácie.

Zisťovanie polohy podľa IP nie je ale úplne spoľahlivé, preto ak nemá užívateľ povolené polohové služby aplikácia nedosiela dáta na server.

Odosielanie dát na server je možné upraviť napríklad zachytením dát cez proxy server a ich upravením, preto server overuje odoslané parametre podobnými technikami. Ak sa vyskytne nejaká nezrovnalosť označí tieto dáta a užívateľom nie sú odoslané do ich aplikácie.

Aplikácia mala najčastejšie problém so zamrznutím práve pri zisťovaní údajov o krajine a načítavaní filtru do mapy. Užívateľia teda po kliknutí na kartu mapy na pomalom pripojení museli chvíľu čakať kým sa mapa otvorila. Vyriešil som to presunutím sieťovej komunikácie na iné procesorové vlákno.

4.3 Tvorba dizajnového návrhu aplikácie

Dizajn aplikácie sa dá riešiť pomocou Interface Builderu, kódom, alebo ich kombináciou.

Interface Builder je súčasť Xcode. Je to grafické prostredie na tvorbu návrhu aplikácie. Objektom vložením do návrhu priradíme triedu pomocou ktorej potom vieme ovplyvňovať zobrazený výsledok. Každý takýto objekt sa môže prepojiť s kódom pomocou IBOutlet v prípade ak chceme meniť jeho vlastnosti (text, farbu, obsah a pod.) alebo pomocou IBAction ak chceme zaznamenať, že užívateľ interagoval s daným objektom (kliknutie, presunutie, dlhý dotyk).

Od verzie Xcode 6 je podpora technológie Auto Layout a Size Classes. Size Classes slúži pre univerzálnu tvorbu aplikácie na všetky druhy rozlíšení a pomerov strán, ktoré Apple ponúka s iOS systémom.

Aplikácia je rozdelená do troch logických obrazoviek, Test, History, Map. Ich popis je uvedený nižšie.

4.3.1 Auto Layout

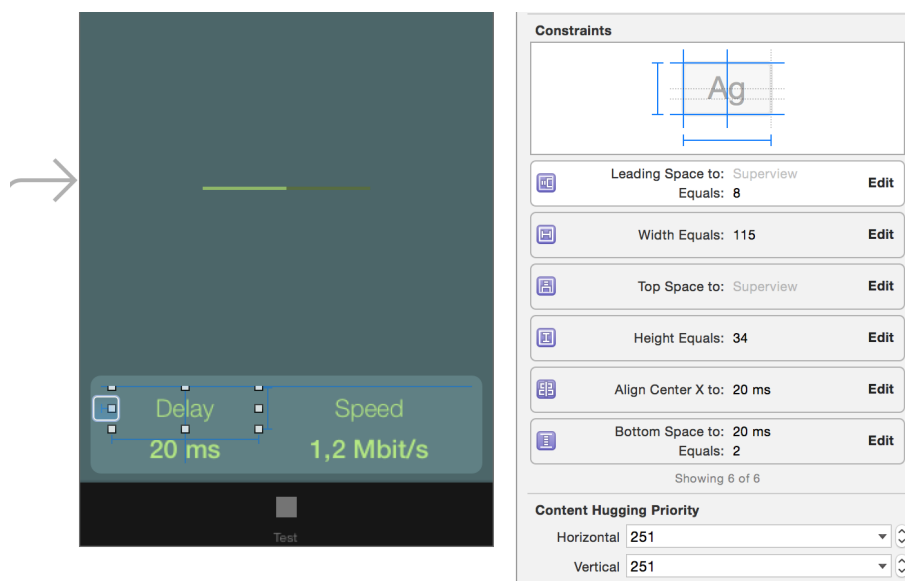
Auto Layout pomáha vývojárom vytvoriť dizajn aplikácie tak, aby sa prispôboval rôznym druhom obrazoviek, módom zobrazenia ako je vodorovné alebo zvislé a tiež dynamickému prispôbovaniu sa rozmerom prvkov podľa aktuálneho obsahu na obrazovke. Pomáhajú tomu constraints pomocou ktorých sa dá definovať na akej pozícii, alebo v akom vzťahu sa bude daný prvok nachádzať od okraja, alebo od iného objektu. Umožňujú priradiť aj prioritu, a tým uprednostniť úpravu rozmeru iného objektu na úkor druhého.

V prípade využívania Auto Layout funkcie je nutné uplatniť constraints na všetky objekty aby Xcode vedel v každom stave, kde sa objekt bude nachádzať. Existuje aj tzv. vizuálny programovací jazyk, ktorý umožňuje nastavovanie constraints kódom. Ak je potrebné meniť za chodu niektorý parameter constraint, ako je vzdialenosť prvkov od seba, je to možné pomocou kódu. Takýto constraint musí byť prepojený pomocou IBOutlet. Tak sa dá docieľiť napríklad animácia posúvania objektov po obrazovke.

Pri vývoji aplikácie som navrhol prostredie pomocou tohoto nástroja. Aplikácia sa teda prispôbuje rôznym obrazovkám od iPhone 4 (3,5") po iPhone 6 Plus (5,5") a vždy využíva efektívne celú plochu obrazovky.

4.3.2 Size Classes

Size Classes je funkcia vývojárskeho nástroja Xcode 6 podporovaná od operačného systému iOS 7, ktorá umožňuje pripraviť obrazovku pre všetky typy obrazoviek, teda



Obr. 4.6: Funkcia constraints

pre iPhone a iPad. Obráz sa potom zväčšuje alebo zmenšuje podľa toho, aký typ zariadenia užívateľ má a nedochádza k efektu rozťahovacia grafiky. Aby to fungovalo dobre je vhodné využiť zároveň funkcie Auto Layout. Jednotlivé Constraints pre každý prvok sa potom dá nastaviť individuálne pre každý typ obrazovky.

Size Classes zároveň umožňuje nastaviť jednotlivé parametre vzdialeností, veľkostí a podobne pre rôzne typy obrazoviek. Takto som mohol optimalizovať zobrazenie pre iPhone a iPad tak, aby bola efektívne využívaná plocha jednotlivých zariadení. Je možné nastaviť chovanie obrazu aj pri režime na ležato, to ale moja aplikácia neumožňuje. Ukážka je zobrazená na obrázku č. 4.7. Vďaka tejto funkcií som vytvoril iPad verziu až na konci celého vývoja iba jej povolením a drobnými úpravami chovania pre niektoré obrazovky aplikácie.

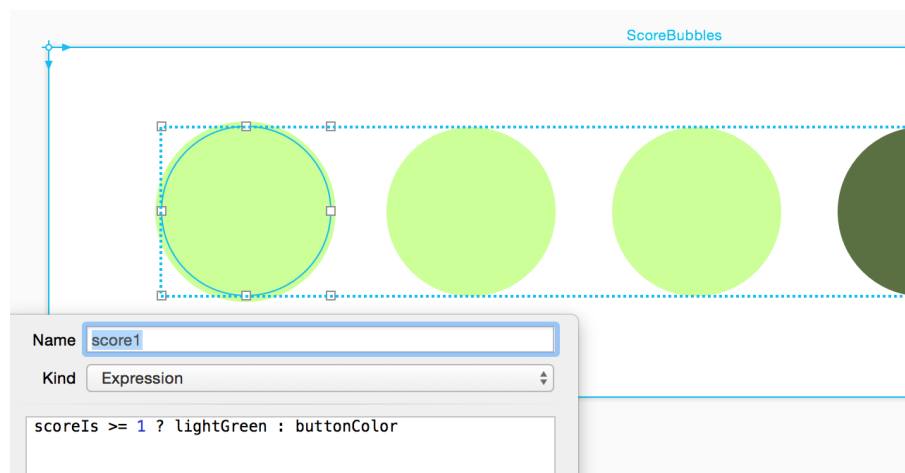
4.3.3 Tvorba objektov a nastavenie grafického prostredia

Aby som zaistil, že objekty budú pri každej veľkosti obrazovky vždy pekne vykreslené využil som nástroja PaintCode, ktorý je z časti grafický editor generujúci kreslené objekty priamo do kódu pomocou bezierových kriviek. Po vložení takéhoto objektu do rámčeka sa tento objekt v aplikácii prispôbuje veľkosti UIView objektu v ktorom je umiestnený. To má výhodu hlavne pri rôznych rozlíšeníach obrazoviek a ich uhlopriečok. Využil som ho ale na viac funkcií, a to:

- Generovanie objektov v režime kresliacej metódy. Objekt nie je takto vykreslený ako obrázok, ale vykresľuje sa počas behu aplikácie. V PaintCode je možné nastaviť chovanie objektov na základe parametru pomocou jednodu-



Obr. 4.7: Nastavovanie chovanie obrazovky pomocou Size Classes



Obr. 4.8: Nastavenie chovania grafického objektu v PaintCode

chého skriptovacie jazyka. Tento parameter je vložená do kódu ako parameter metódy, ktorá tento objekt vykresľuje. V aplikácii sa potom volá metóda na vykreslenie objektu s daným parametrom a objekt sa pomocou neho vykreslí. Ukážka je zobrazená na obrázku č. 4.8

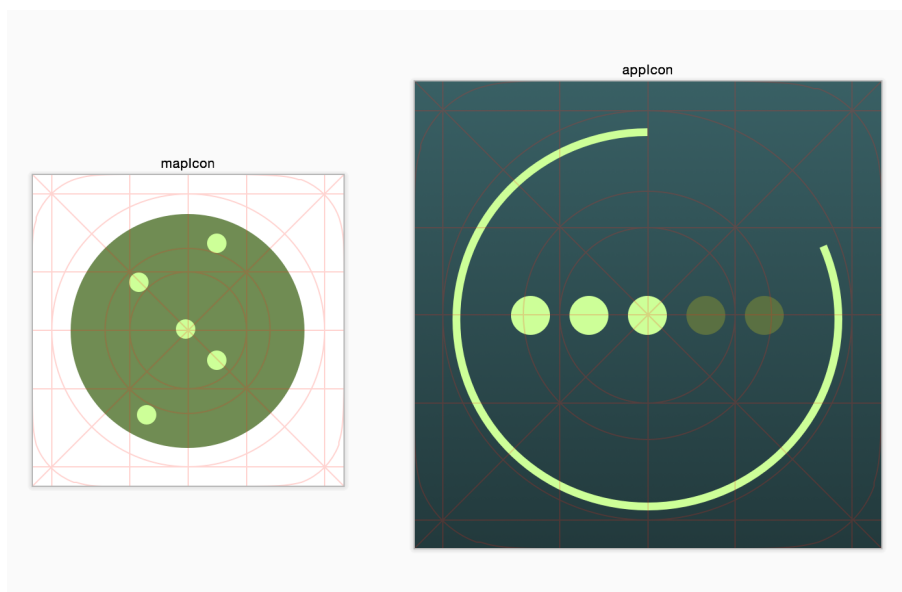
- Pomocou PaintCode som riešil prácu s farbami použitých v aplikácii. Vytvoril som jednu referenčnú farbu a ostatné som definoval na základe nej, napríklad zvýšením jej jasů.
- Vytvoril som v ňom aj ikony k aplikácii, ktoré automaticky vygeneroval do všetkých potrebných veľkostí, ktoré aplikácia potrebuje. Návrh je na obrázku č. 4.9

4.3.4 Obrazovka Test

Obrazovka Test zobrazená na obrázku č. 4.10 je prvá, ktorú užívateľ uvidí po spustení aplikácie. Má za účel umožniť spustiť meranie na ktorého konci sa užívateľ dozvie aké kvalitné ma pripojenie. Umožňuje to tlačítko umiestnené uprostred obrazovky. Po jeho stlačení sa spustí prvá časť merania, a to meranie doby HTTP odpovede na požiadavok o načítanie stránky. Meranie prebieha 5x a užívateľ je o priebehu informovaný pomocou indikátor priebehu a zároveň sa mu zobrazujú výsledné hodnoty. Po skončení tohoto merania sa spustí meranie rýchlosti internetu. Trvá 10 sekúnd a užívateľ je o stave informovaný kruhovým indikátorom priebehu.

4.3.5 Obrazovka My History

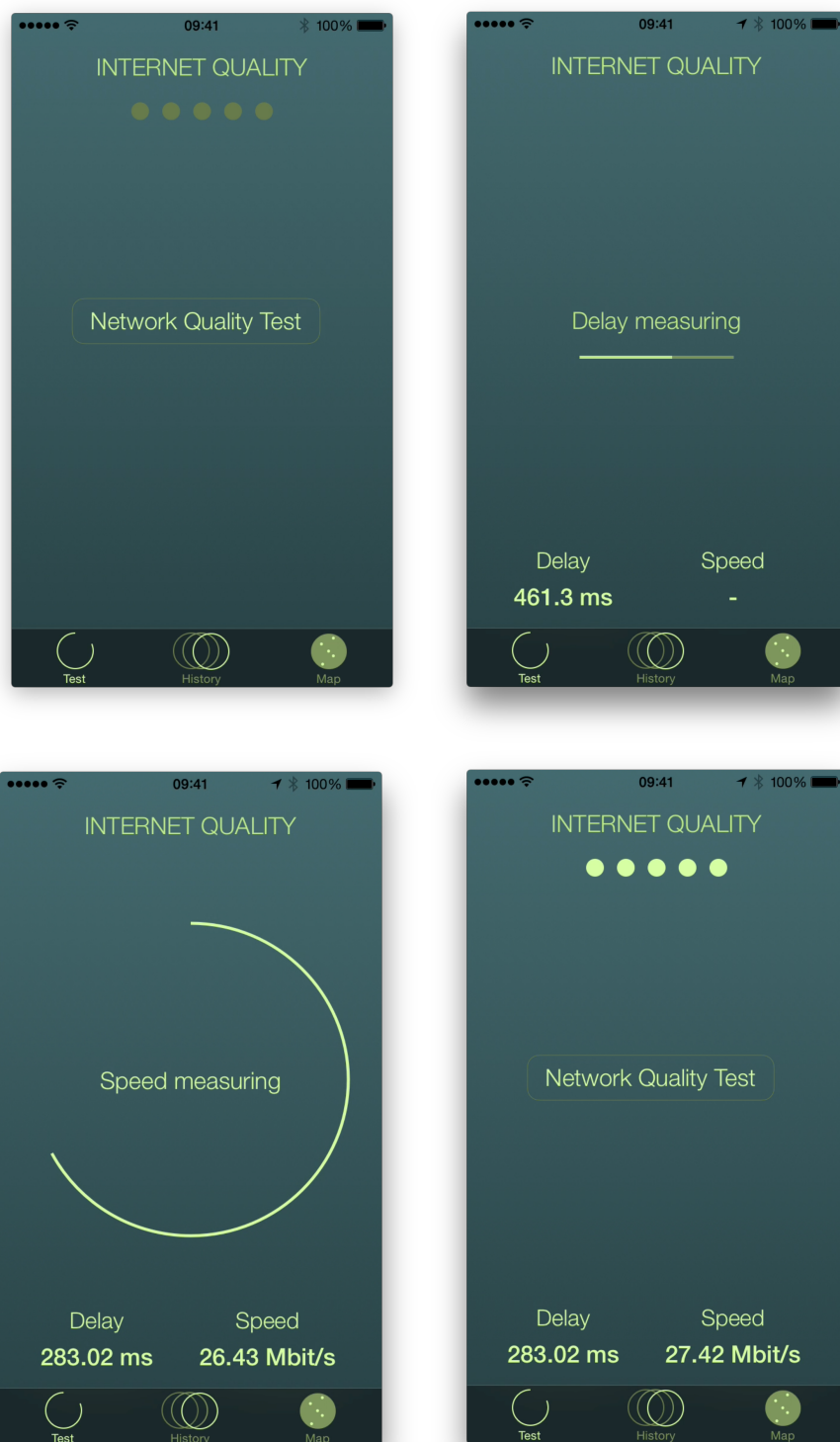
Obrazovka My History zobrazuje históriu všetkých meraní, ktoré užívateľ vykonal na svojom zariadení obr. č. 4.11. Je rozdelená na dve časti. V prvej je zoznam, ktorá



Obr. 4.9: Návrh ikon pre aplikáciu

zoradí merania podľa dátumu vykonania testu. Každý test obsahuje na tejto obrazovke základne hodnoty k meraniu a po kliknutí naň sa zobrazí detail merania. Ten obsahuje navyše miesto na mape, kde bolo meranie vykonané, graf priebehu merania prenosovej rýchlosti a detailný popis celého merania obr. č. 4.12. Informácie na oboch obrazovkách sa dynamicky menia podľa vlastností merania, ako to, či bolo meranie robené na WiFi alebo mobilnom pripojení. Otváranie detailu merania je možné z troch miest v aplikácii, a to ikonou po ukončení merania, kedy sa zobrazí detail posledného merania. Druhou možnosťou je ho otvoriť cez obrazovku My History a poslednou je priamo z mapy. Ak užívateľ otvorí túto obrazovku práve posledným spôsobom, nezobrazujú sa všetky dáta tak, ako pri jeho vlastných meraniach. Vynechaný je názov WiFi siete na ktorej bolo meranie vykonané a počet dát ktoré boli pre toto meranie spotrebované. Tieto informácie by mohli narušovať súkromie užívateľov a nie sú ani odosielané na server.

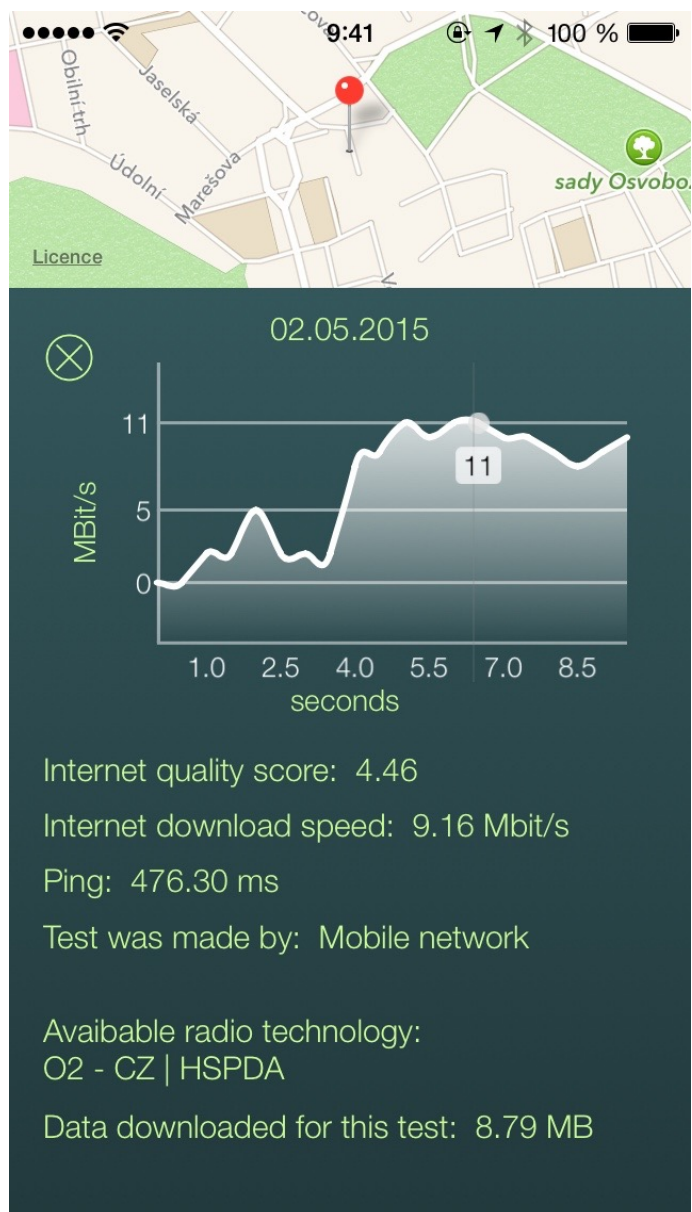
Aby sa obrazovka mohla dynamicky prispôbovať parametrom ktoré prijíme, využil som v možnosti Auto Layout a priorit. Príklad je na obr. č. 4.13 kde je zobrazené, že label zobrazujúci informáciu o počtu stiahnutých dát môže byť vzdialených od iných labelov rôzne veľkosti. Prerušovaná čiara znamená, že táto vzdialenosť má nastavenú nejakú prioritu menšiu ako 1000 (1000 znamená prioritu maximálnu, teda vyžadovanú). Pomocou kódu môžu byť niektoré label nad ním odstránené a vtedy sa uprednostní vzdialenosť s ďalšou najvyššou prioritou.



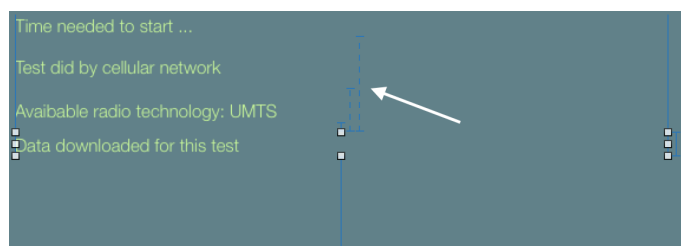
Obr. 4.10: Pribeh obrazovky Test



Obr. 4.11: Obrazovka My History



Obr. 4.12: Obrazovka My History Detail



Obr. 4.13: Nastavovanie priorít pomocou Auto Layout

4.3.6 Obrazovka Map

Obrazovka Mapa má za úlohu zobrazit užívateľovi namerané hodnoty všetkých užívateľov z polohy, ktorú ma zobrazenú na obrazovke zariadenia. Užívateľ si môže nastaviť rôzne filtre, ktorých obsah je načítaný buď z online databáze alebo z lokálnej pomocou Core Data. Tieto filtre sú zobrazené na obrázku č. 4.16 a sú to:

- Jednotlivý operátori pôsobiaci v krajine kde sa užívateľ nachádza
- Rádiove technológie poskytujúci tieto operátori
- Na akom pripojení bolo meranie vykonané (mobilné, wifi)
- Online databáza alebo lokálna s dátami meraného priamo užívateľom

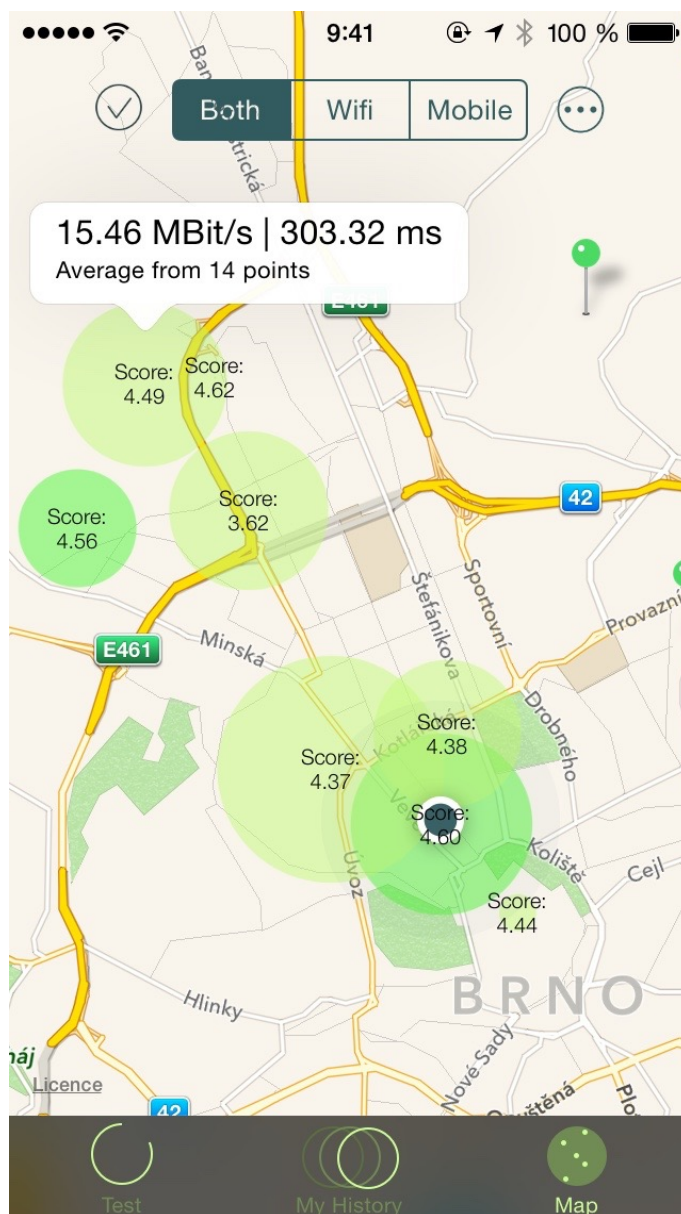
Mapa využíva algoritmus klasteringu, ktorý zlučuje jednotlivé body merania do jedného v prípade, ak sa ich na obrazovke na jednom mieste zobrazí príliš veľa. Aplikácia z takto zlúčených bodov vypočíta medián hodnôt QoE, rýchlosti sťahovania a oneskorenia. Na základe tejto hodnoty QoE zobrazí oblasť z ktorej boli jednotlivé body zlúčené do farebného priehľadného kruhu. Jeho farba sa odvíja od hodnoty QoE, od červenej pre hodnotu 1 až po zelenú pre hodnotu 5. Užívatelia tak môžu vidieť po oddialení mapy kvalitu internetového pripojenia pre jednotlivé oblasti graficky tak, ako je zobrazené na obrázku č. 4.14.

Užívateľ má možnosť otvoriť si detail jednotlivého merania na podobne ako pri otvorení detailu vlastných meraní. Pri tomto požiadavku sa dostávajú dodatočné potrebné informácie na vykreslenie grafu a ďalších detailov verejne prístupné z online databáze.

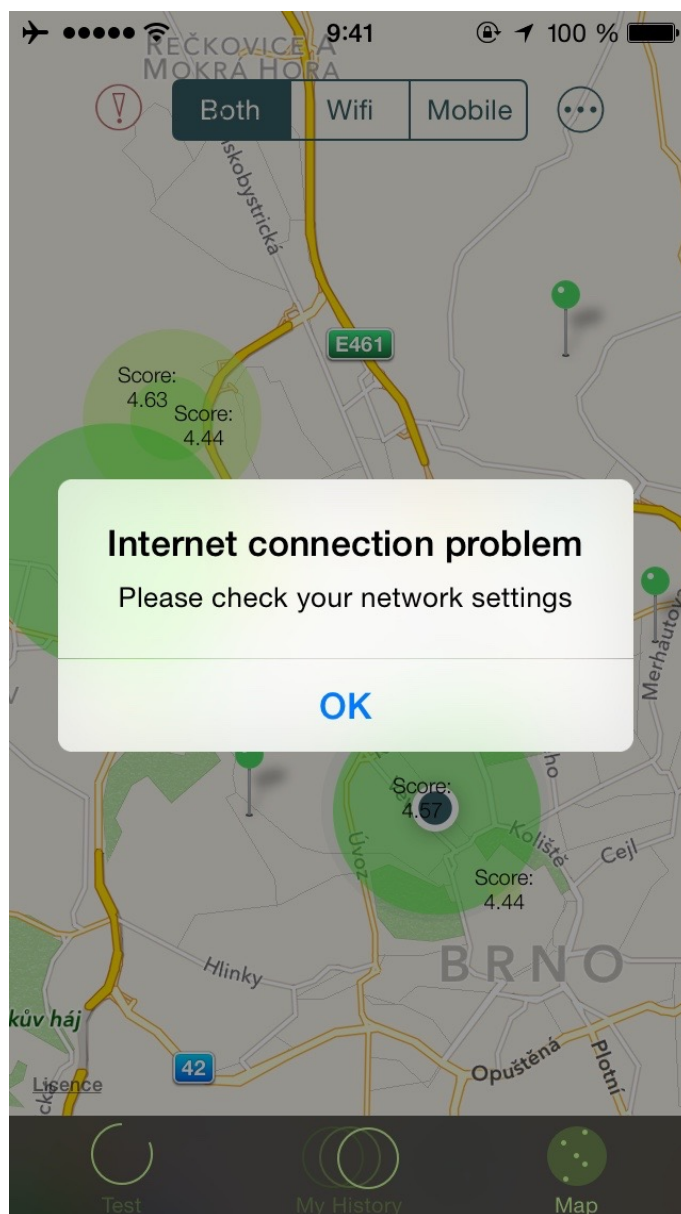
Mimo iné je na tejto obrazovke aj indikátor stavu stiahnutia dát. Ak sa užívateľovi nepodarí stiahnuť dáta zo serveru pre jeho problémy s pripojením, alebo s iným problémom pri sťahovaní dát zo serveru, zobrazí sa červená ikona signalizujúca problém. Ak na ňu užívateľ klikne, zobrazia sa podrobnosti problému pomocou objektu Alert View, tak ako je zobrazené na obrázku č. 4.15. Uprednostnil som zobrazovanie problému ikonou, pretože pri častom sťahovaní dát sa sťahovanie určitého bloku dát nie vždy podarilo. Takto teda užívateľ nemusí neustále potvrdzovať problém so spojením a zároveň môže pracovať s mapou aj keď sa nové dáta nestahujú nerušene.

4.4 Práca s dátami z meraní

Mobilná aplikácia má za účel nie len informovať užívateľa o výsledku jeho merania, ale aj umožniť mu porovnať jeho výsledky s hodnotami ostatných užívateľov. Pre tieto účely som vytvoril tabuľku v databázovom serveri založenom na technológii MySQL. Výhoda použitia tejto technológie je rýchlosť spracovania dát, jej jednoduchého napojenia rôznych aplikácií a širokej základne používateľov a teda i jej



Obr. 4.14: Obrazovka mapa



Obr. 4.15: Obrazovka mapa error hláška

podpora.

V aplikácií som vytvoril lokálnu databázu pomocou technológie Core Data. Tá slúži k zobrazovaniu histórie meraní užívateľa, a to bez nutnosti sťahovať dáta z databázového serveru cez internet.

Užívateľ si vo filtre mapy ma možnosť vybrať online databázu meraní alebo jeho lokálnu. Aplikácia potom pracuje online alebo s Core Data.

4.4.1 Online databáza

Prepojenie databáze a iných služieb je najčastejšie realizované pomocou PHP skriptov, ktoré sú umiestnené na rovnakom serveri ako je databáza. Hlavné výhody tohoto riešenia sú:

- Bezpečnosť
 - Prístup k databáze majú len aplikácie bežiacie na rovnakom serveri
 - Prístupové dáta, ako je užívateľské meno, heslo, názov tabuľky a pod. nie sú distribuované užívateľom v ich aplikáciach
- Pri aktualizácii prístupových dát k databáze stačí aktualizovať parametre v skripte na serveri a nie je nutné distribuovať nové nastavenia všetkým užívateľom do ich klientskej aplikácie
- Skript môže spracovávať dáta, a tak šetriť zdroje klientských aplikácií, ako je výpočtový výkon, spotreba dát
- V závislosti na vytvorení modelu v klientskej aplikácii je možné meniť chovanie pre užívateľov zmenou parametrov skriptov

Aplikácia komunikuje so serverom vždy keď získa nové dáta z merania, alebo ak nejaké dáta potrebuje získať. Vždy sa spolu s požiadavkom o dáta odošlú aj dodatočné informácie, ktoré slúžia k zabezpečeniu korektného formátu. Ide napríklad o verziu aplikácie, a tak je možné upraviť chovanie serveru na požiadavok pre rôzne klientské verzie.

Odosielanie dát po skončení merania je vykonávané na pozadí a užívateľ o tom nie je nijak informovaný. Aplikácia ani nerieši stav zlyhania tohoto odosielania.

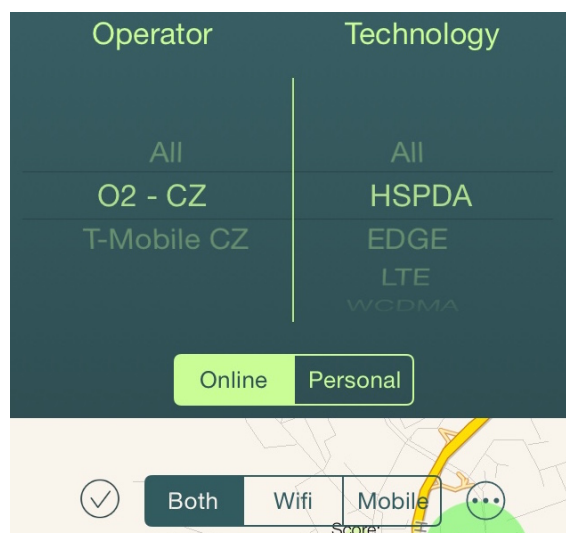
V tabuľke č. 4.4.1 sú uvedené všetky parametre, ktoré aplikácia odosiela do databázy.

Naopak pri prijímaní dát, napríklad bodov merania do obrazovky s mapou, je užívateľ informovaný o stave spojenia, a to načítavanie, korektného stiahnutia dát prípadne problému.

Mapový kontrolér si po jeho prvom otvorení v aplikácii vyžiada zo servera všetkých operátorov a rádiové technológie dostupné v krajine, v ktorej sa užívateľ nachádza. Po ich načítaní aplikácia požiada server zavolaním príslušného skriptu s parametrami všetkých operátorov, dostupných rádiových technológií a GPS súradníc pre

Tab. 4.1: Parametre odosielané na server

Parameter	Význam
Rýchlosť sťahovania	Priemerná rýchlosť sťahovania v MBit/s
Oneskorenie HTTP odpovede	Priemerná doba oneskorenia na HTTP požiadavok v ms
QoE	Vypočítaná hodnota parametru QoE
Prístup užívateľa k internetu	Overenie, či bolo meranie uskutočnené pomocou pripojenie WiFi, alebo mobilnej dátovej siete WWAN
Dostupná rádiová technológia WWAN	K akej mobilnej dátovej technológii je zariadenie pripojené (napr. EDGE, HSPDA, LTE...)
RAW dáta	Nespracované hodnoty z merania, ktoré reprezentujú prenosovú rýchlosť v pravidelných intervaloch počas merania
Operátor	Meno operátora vlastniaceho sieť ku ktorej je zariadenie pripojené
ISO kód zeme	Určuje v ktorej krajine operátor pôsobí
Zemepisná dĺžka	GPS súradnice užívateľa v dobe merania
Zemepisná šírka	
Identifikačný kód aplikácie	Jedinečný kód pomocou ktorého je možné identifikovať užívateľa
Meno operačného systému	Identifikácia operačného systému
Verzia operačného systému	
Model telefónu	
Detailne model telefónu	Určuje o aký konkrétny model telefónu ide



Obr. 4.16: Filtre pre mapu

minimálnu a maximálnu zemepisnú šírku a dĺžku ktorú má užívateľ zobrazenú na obrazovke. Aplikácia potom čaká na odpoveď. Ak príde v poriadku, delegovaná metóda objektu, ktorý komunikáciu, stahovanie dát a úpravu JSON odpovede do vhodného formátu, odošle spracované dáta späť do mapového kontroléru. Maximálny počet ktorý bodov merania je limitovaný priamo na serveri a odosielaajú sa iba najnovšie dáta.

Tieto dáta, ktoré obsahujú informácie ako je poloha merania, prenosová rýchlosť, oneskorenie, QoE a podobne sa vložia do objektu KPCLusteringController ktorý zabezpečuje zlučovanie jednotlivých bodov do jedného v prípade potreby. Vďaka tomuto zlučovaniu sa navyše výrazne znížili nároky na výpočtový výkon zariadenia, keďže nemusia vykresľovať veľký počet bodov na mape. Všetky body merania sa ešte spracovávajú podľa toho, či je daný bod zlúčený z viacerých, alebo ide o samostatný bod merania.

Pre zlúčené sa vypočíta medián prenosovej rýchlosti, oneskorenia a parametru QoE, ktoré sú potom zobrazené v anotácii pre kliknutie na daný bod. Bod sa zobrazuje vo forme farbeného kruhu. Znova som uprednostnil medián aby sa prípadne extrémne hodnoty výrazne neprejavili vo výsledných hodnotách.

Anotácia bodu reprezentujúceho jednotlivé meranie obsahuje navyše tlačítko po ktorého kliknutí sa zobrazí obrazovka detailu merania. Na základe parametru id sa odošle požiadavka na server pre vrátenie dodatočných informácií potrebných na zobrazenie grafu a pod. Ak je požiadavka z nejakého dôvodu nesplnená užívateľ je o tom informovaný pomocou Alert View a okno detailu merania sa automaticky zavrie.

ENTITIES		Attributes	
E MeasuredData		Attribute ^	Type
FETCH REQUESTS		N accuracy	Double
CONFIGURATIONS		D date	Date
C Default		N downloadedData	Double
		N downloadSpeed	Double
		N latency	Double
		N latitude	Double
		N longlat	Double
		S radioTechnology	String
		S reachableVia	String
		N rscp	Integer 16
		+ -	

Obr. 4.17: Návrh modelu pre Core Data

4.4.2 Core Data

Core data slúžia k návrhu modelu pomocou tzv. entít, atribútov a vzťahmi medzi nimi. Sú optimalizované k efektívnemu využívaniu výkonu zariadenia, ako aj pamäti, a k rýchlemu prístupu k dátam v nich uložených.

Moja aplikácia využíva iba ich základné možnosti. Vytvorená entita MeasuredData, reprezentuje model, alebo šablonu, do ktorej sa ukladajú dáta do atribútov tak, ako sú zobrazené na obrázku 4.17

Pre komunikáciu s Core Data sa používa objekt `NSManagedObjectContext` a metóda `managedObjectContext`, ktoré sprostredkovávajú komunikáciu s Core Data.

```
//Nastavenie prístupu ku Core Data
id delegate = [[UIApplication sharedApplication]delegate];
NSManagedObjectContext *context = [delegate managedObjectContext];
```

```
//Pridanie hodnoty do atribútu entity
//Vyber správnej entity
NSManagedObject *measuredData = [NSEntityDescription insertNewObjectForEntityForName:@"MeasuredData"
inManagedObjectContext:context];
[measuredData setValue:[NSNumber numberWithInt:accuracy] forKey:@"accuracy"];
```

```
//Čítanie hodnoty z vybranej entity
NSFetchRequest *fetchRequest = [NSFetchRequest fetchRequestWithEntityName:@"MeasuredData"];
fetchRequest.sortDescriptors = @[[NSSortDescriptor sortDescriptorWithKey:@"date" ascending:NO]];
[measuredData setValue:[NSNumber numberWithInt:accuracy] forKey:@"accuracy"];

NSArray *fetchedMeasure = [context executeFetchRequest:fetchRequest error:&error];
for (NSManagedObject *info in fetchedMeasure) {
[accuracy setValue:[info valueForKey:@"accuracy"] forKey:@"accuracy"];
}
```

```
//Nastavenie filtru pre získanie dát podľa užívateľského nastavenia
[fetchRequest setPredicate:[NSPredicate predicateWithFormat:@"radioTechnology == %@", technologies]];
```

PHP skripty

PHP skripty sú nastavené na prijímanie dát pomocou POST metódy. Narozdiel od GET metódy nie je potrebné využívať URN časť HTML adresy, ktorá je obmedzená dĺžkou.

Pri ukladaní dát do databázy sa po volaní príslušnej php stránky s príslušnými parametrami, ktoré sú zobrazené v tabuľke č. 4.4.1 a skontroluje, či sú všetky požadované hodnoty správne. Tie sa potom odošlú do databázy príkazom `Insert INTO` do príslušnej tabuľky databázy.

Odosielanie dát na server je možné upraviť napríklad zachytením dát cez proxy server a ich upravením, preto server overuje odoslané parametre pomocou ďalších metód. Ak sa vyskytne nejaká nezrovnalosť označí tieto dáta a užívateľom nie sú odoslané do ich aplikácie.

4.5 Distribúcia aplikácie

Pri vývoji aplikácie som potreboval získať užívateľov, ktorý by aplikáciu otestovali. Týmto spôsobom som mohol odhaliť hlavné problémy, ktoré sa objavili až pri rôznych konfiguráciách zariadení, chovaní užívateľov a podmienok v rôznych krajinách na svete pre mobilný internet. Zároveň som implementoval do aplikácie monitorovacie mechanizmy, ktoré ma upozornili na chyby v aplikácií, ktoré sa vyskytli u užívateľov a tiež sledovanie chovania užívateľov v aplikácií.

4.5.1 Formy distribúcie iOS aplikácie medzi beta testerov

Aplikácia, ktorá sa nenachádza v App store sa dá nahráť do Apple zariadený spôsobmi:

- Registrovaným Apple zariadenia ako vývojárske na developerskom portáli <https://developer.apple.com> kde sa potom musí pre príslušný počítač Mac stiahnuť Provision profile. Ten je možné stiahnuť iba s platným vývojárskym účtom. Potom je možné projekt v Xcode preložiť a nahráť priamo na zariadenie.
- Distribuovať aplikáciu pomocou korporátneho účtu ako aplikáciu, ktorú je možné umiestniť na vlastné servery.
- Od verzie operačného systému iOS 8 Apple umožnil vývojárom beta testovanie aplikácie pomocou programu Test Flight. Ten je rozdelený na interných a externých testov. Interní testeria môžu získať aplikáciu podobne ako je popísané v prvom bode, ale nie je nutné nahrávať ju do zariadenia cez Mac, ale cez program Test Flight. Pre externých testerov je nutné poslať aplikáciu na

schválenie do Apple, ktorý ju skontroluje a ak je všetko v poriadku, umožní rozoslať aplikáciu medzi maximálne 1000 užívateľov. Tí, ak príjmu pozvánku do programu testovania, si môžu nainštalovať aplikáciu do svojich zariadení. Následne sú upozorňovaní na prípadne aktualizácie.

Aplikáciu som pridal do programu Test Flight a požiadal som o schválenie do verejného beta programu. Schvaľovanie trvalo 3 dni po ktorom som začal rozosielať pozvánku. Podarilo sa mi získať 40 testerov z 13 krajín sveta a celkovo som im rozoslal 10 verzií na testovanie. Počas testovania títo testerí spravili niekoľko stoviek meraní. Vďaka globálnemu testovaniu som odhalil problém s niektorými operátormi, ktorý nevracajú správne meno operátora, problém roamingových užívateľov, určeným polohy zariadenia a tiež niektoré nejasnosti po stránke dizajnu a jeho problému. Testovanie medzi užívateľmi výrazne urýchlilo vývoj a odhaľovanie chýb v aplikácií.

Aplikácia je určená aj pre zariadenia s iOS 7, ktorý ale Test Flight nepodporuje. Testoval som to ale aj na tieto zariadenia a to pomocou služby Fabric s ktorou bolo možné distribuovať aplikáciu cez internet. Tieto zariadenia ale museli byť pridané do vývojárskeho programu a musel som vložiť do projektu nový provision profile, ktorý podpisoval aj tieto zariadenia rámci projektu.

4.5.2 Distribúcia aplikácie do App Store

Aby mohla byť aplikácia odoslaná do App store je nutné pripraviť všetku grafiku, ako sú ikony a obrázky aplikácie. Jednotlivé ikony musia byť pre rozlíšenie pre všetky typy zariadení a ich rozlíšenia, čo pre všetky uhlopriečky iPhone a iPad znamená 19 rôznych ikon, tak ako je zobrazené na obrázku č. 4.18. Pomocou Xcode sa vytvorí build, ktorý sa priamo z neho odošle do App Store portálu. Už pri odosielaní sa skontrolujú základné požiadavky na aplikáciu. Ak je všetko v poriadku, cez portál <http://itunesconnect.apple.com> je možné vybrať daný build a odoslať ho na schválenie. Na tej istej stránke sa potom doplnia informácie k aplikácií, ktoré budú zobrazené v popise aplikácie, pridajú sa fotky a ďalšie informácie. Schvaľovanie potom môže trvať aj niekoľko týždňov.

4.5.3 Monitorovanie distribuovanej aplikácie

Aby som mohol odhaľovať chyby v kóde aplikácie, ktoré mohli spôsobiť pád aplikácie využil som služby Fabric.io. Ako druhý monitorovací nástroj je implementovaná do aplikácie služba Mixpanel, ktorá sleduje chovanie užívateľov v aplikácií, prípadne zaznamenáva stavy, ktoré sa vyskytnú v aplikácií.



Obr. 4.18: Ikony aplikácie pre všetky zariadenia

Fabric

Hlavný účel služby je hlásenie chýb, ktoré spôsobia pád aplikácie. Ak takáto situácia nastane, príde správa o tom, na ktorom mieste v kóde došlo k chybe a čo jej predchádzalo. Mimo iné služba monitoruje počet aktívnych užívateľov v reálnom čase, ako dlho užívatelia používajú aplikáciu a na aké operačné systémy a zariadenia používajú. Umožňuje aj vyššie popísanú distribúciu aplikácie. Ukáža je zobrazená na obrázku č. 4.19

Mixpanel

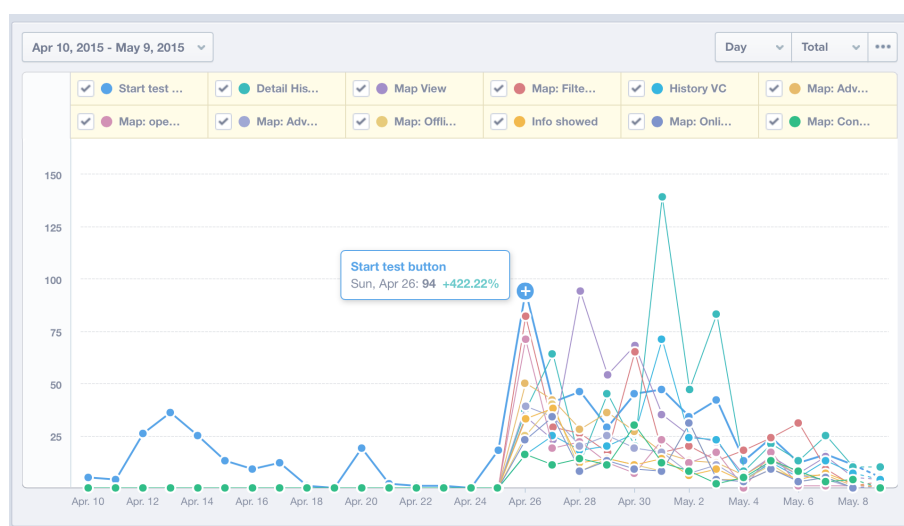
Pomocou tejto služby som mohol jednoducho implementovať zaznamenávanie chovania užívateľov v aplikácií. Pri udalostiach ktoré som chcel zaznamenávať som pridal nasledujúci kód:

```
Mixpanel *mixpanel = [Mixpanel sharedInstance];
[mixpanel track:@"Nazov udalosti"];
[mixpanel flush];
```

Toto sa odošle na servery tejto služby a potom je možné zobrazíť grafy, na čo užívatelia klikali, v akom poradí a pod. Využil som to aj na zaznamenávanie niektorých stavov, ako je nedostupné spojenie so serverom. Grafy zobrazujúce počet kliknutí na stránkach Mixpanel.com sú zobrazené na obrázku č. 4.20.



Obr. 4.19: Ukážka služby Fabric.io



Obr. 4.20: Zaznamenané udalosti pomocou služby mixpanel

5 VYHODNOTENIE UŽÍVATEĽMI NAMERA- NÝCH HODNÔT

Počas celej doby testovania som nazbieral od užívateľov reálne hodnoty ich kvality pripojenia pomocou modelu popísaného v predchádzajúcich kapitolách. Pre Českú republiku ide o 314 meraní z toho 85 na sieťach mobilných operátorov, a to hlavne pre oblasť Brno. Na základe týchto meraní som vytvoril grafy č. 5.1 zobrazujúce hodnotu MOS pre mobilné dátové siete a WiFi siete v Brne. Graf č. 5.2 a č. 5.3 ukazuje koľko percent užívateľov dosahuje akú rýchlosť a oneskorenie. Väčšina užívateľov dosahuje rýchlosť do 10 MBit/s, a to na oboch typoch pripojenia.

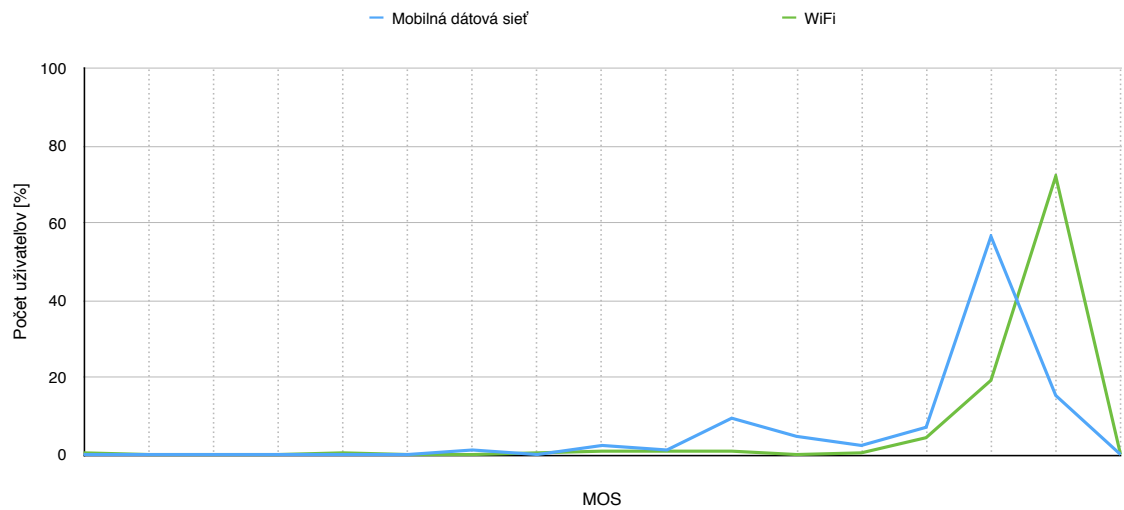
Ďalší graf č. 5.4 zobrazuje, aký vplyv majú jednotlivé rádiové technológie na výsledný MOS. Z grafov vyplýva, že WiFi pripojenie je stále kvalitnejšie ako to mobilné, aj keď pri technológii LTE je už podľa meraní pri 50% prípadov kvalitatívne vyrovnané.

Od užívateľov sa podarilo pre Českú republiku získať dáta najmä od operátorov O2 CZ a T-mobile CZ. Na základe týchto dát som vytvoril graf č. 5.5 ktorý porovnáva priemerne dosiahnuté odhadnuté MOS pre všetky urobené merania na ich sieti. U operátora O2 CZ môže znižovať dosiahnuté hodnoty to, že väčšina užívateľov pravdepodobne nemala podporu LTE zariadení. Naopak T-mobile užívatelia túto technológiu väčšinou využívali.

Pomocou obrazovky v mape je možné zobraziť kvalitu internetu v jednotlivých oblastiach. Pri pohľade na oblasť Brno a okolie je vidno, že internet v meste má lepšie parametre ako v jeho okolí. V meste priemerná rýchlosť sťahovania dosahuje okolo 7 MBit/s. V jeho okolí sú to často hodnoty v kbit/s, tak ako je zobrazené na obrázku č. 5.6.

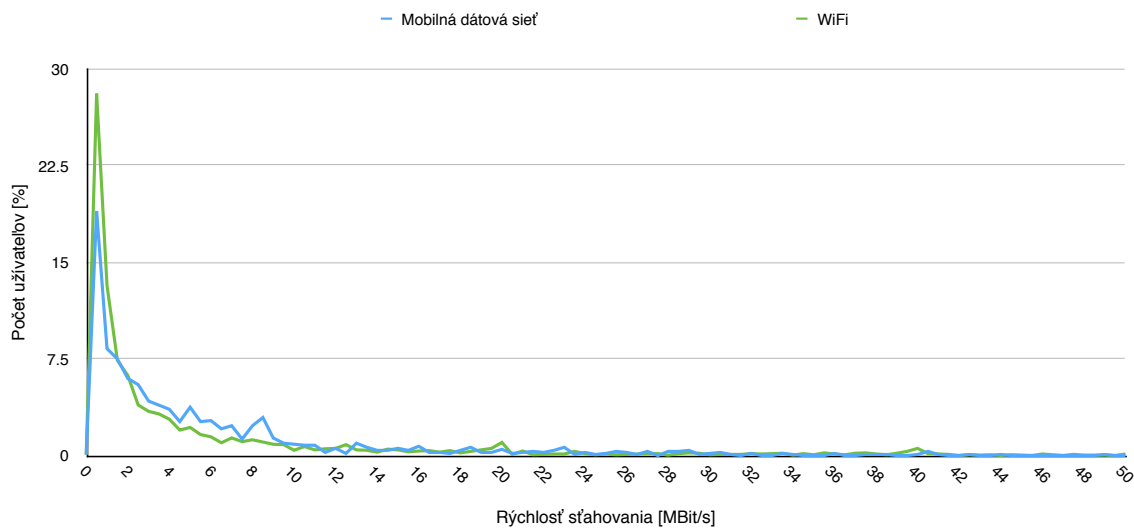
Aplikácia sa po uvedení do App Store rozšírila aj medzi užívateľov po svete a to najmä do krajín USA a Veľkej Británie. Zo všetkých meraní pre tieto krajiny urobených na sieťach operátorov som spravil graf č. 5.7, ktorý ukazuje rozdieli v priemerných hodnotách kvality internetu medzi jednotlivými krajinami. Vo Veľkej Británii mala väčšina užívateľov dostupné LTE pripojenie. Väčšina meraní vo Veľkej Británii bola robená v husto zabývaných oblastiach. Z toho vidno, že LTE technológia je dostatočne kvalitná aj pre tieto oblasti.

V tabuľke č. 5.1 sú zobrazené hodnoty o počte užívateľov, ich zariadení a ďalších podrobnosti, ktoré som počas vývoja a po distribúcií aplikácie nazbieral. Aplikácia je podľa štatistík stabilná na zariadeniach užívateľov, pretože k pádu aplikácie došlo iba u 0,2% užívateľov. Najviac stiahnutí aplikácie bolo v krajinách USA, a to pravdepodobne z dôvodu najväčšieho rozšírenia platformy iOS na tomto trhu. Priemerná rýchlosť medzi WiFi a mobilnou sieťou sa približuje k sebe a sú veľmi podobné, čo

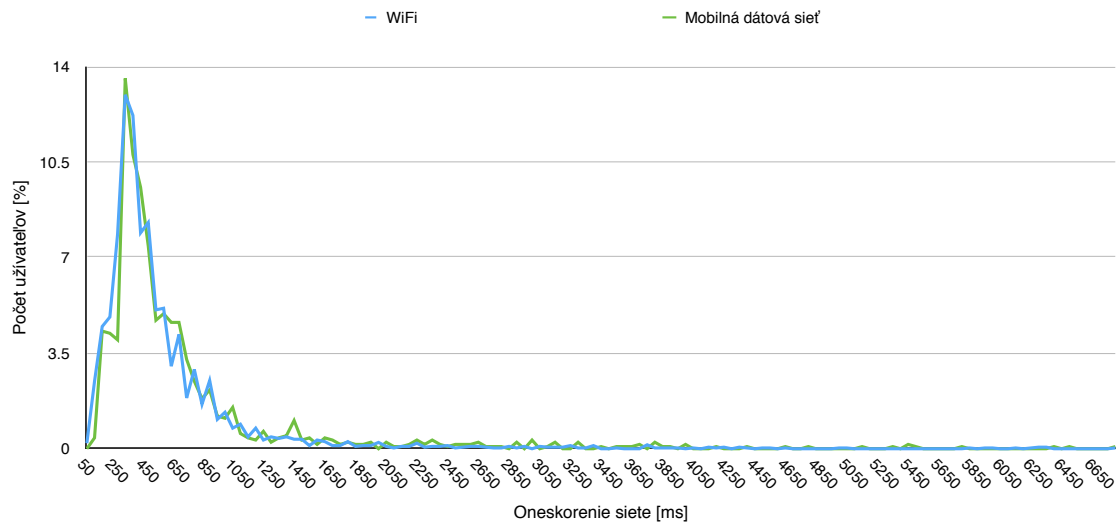


Obr. 5.1: Porovnanie parametru MOS pre mobilnú dátovú sieť a WiFi

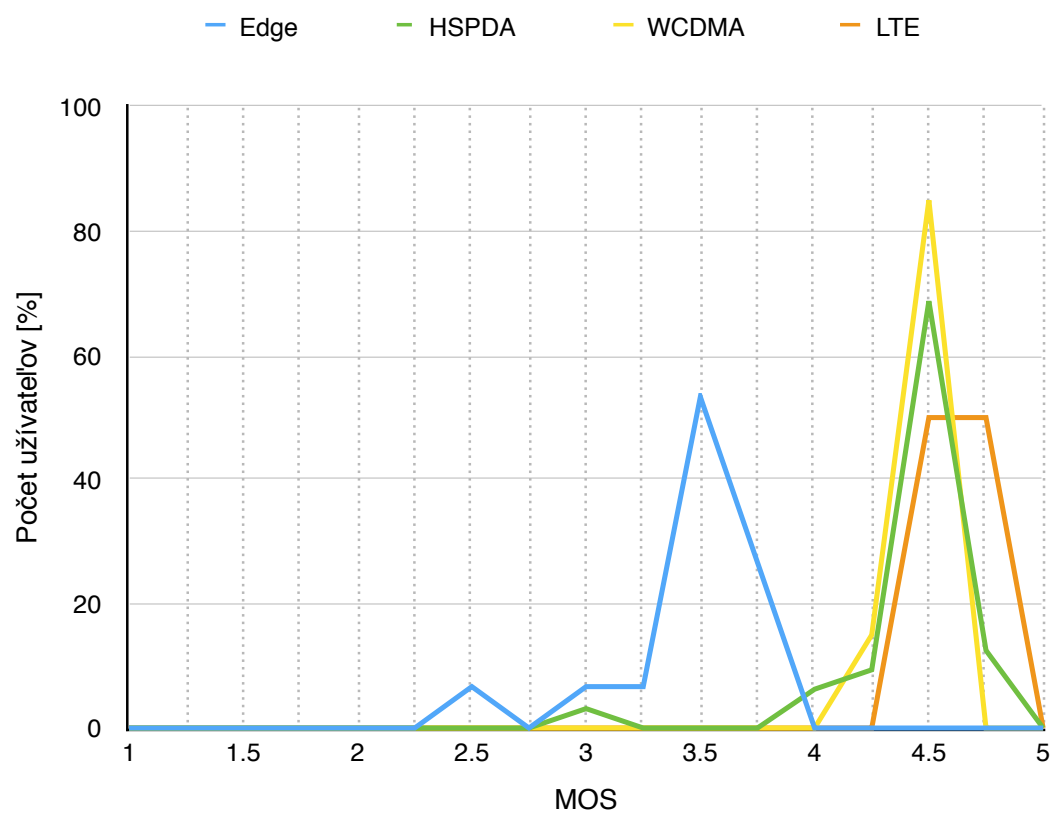
je spôsobené hlavne rozšírením pokrytia LTE, na ktorom bolo urobených najviac testov na mobilnom pripojení.



Obr. 5.2: Porovnanie prenosovej rýchlosti pre mobilnú dátovú sieť a WiFi



Obr. 5.3: Porovnanie oneskorenia siete pre mobilnú dátovú sieť a WiFi



Obr. 5.4: Porovnanie vplyvu jednotlivých rádiových technológií na MOS

Tab. 5.1: Zhrnutie dát o aplikácií

Počet zariadení

Typ zariadenia	Počet
iPad	122
iPhone	4538

Počet beta testerov	Počet krajín v ktorých sa aplikácia použila	Užívatelia bez zaznamenaného pádu aplikácie
36	15	99,8%

Počet meraní na mobilnom pripojení

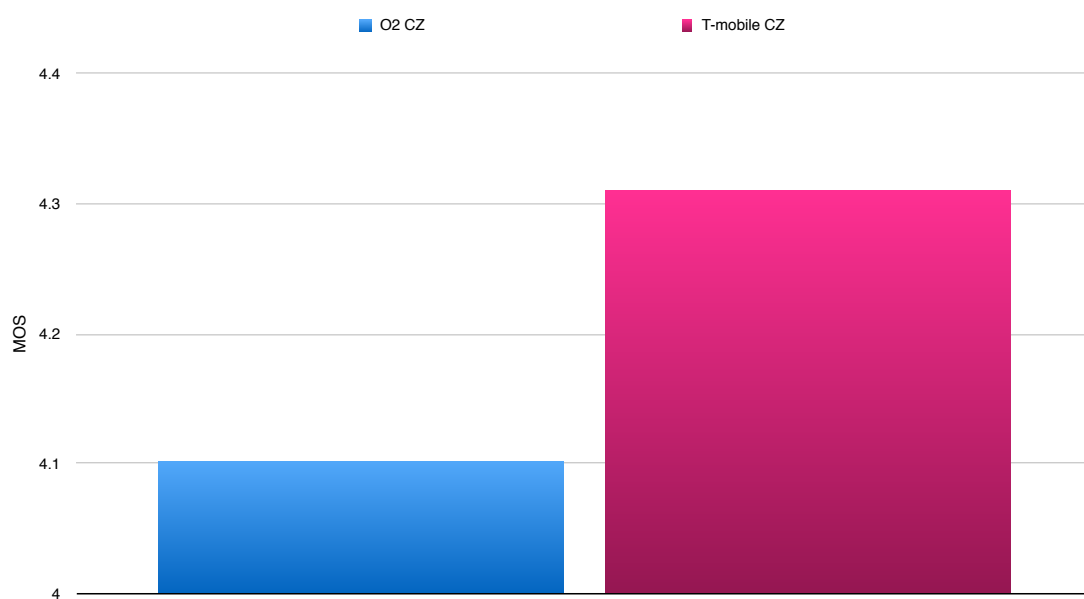
Celkovo	Edge	HRPD	HSPDA	HSUPA	WCDMA	LTE
1253	52	2	406	34	230	529

Maximálne a priemerné rýchlosti

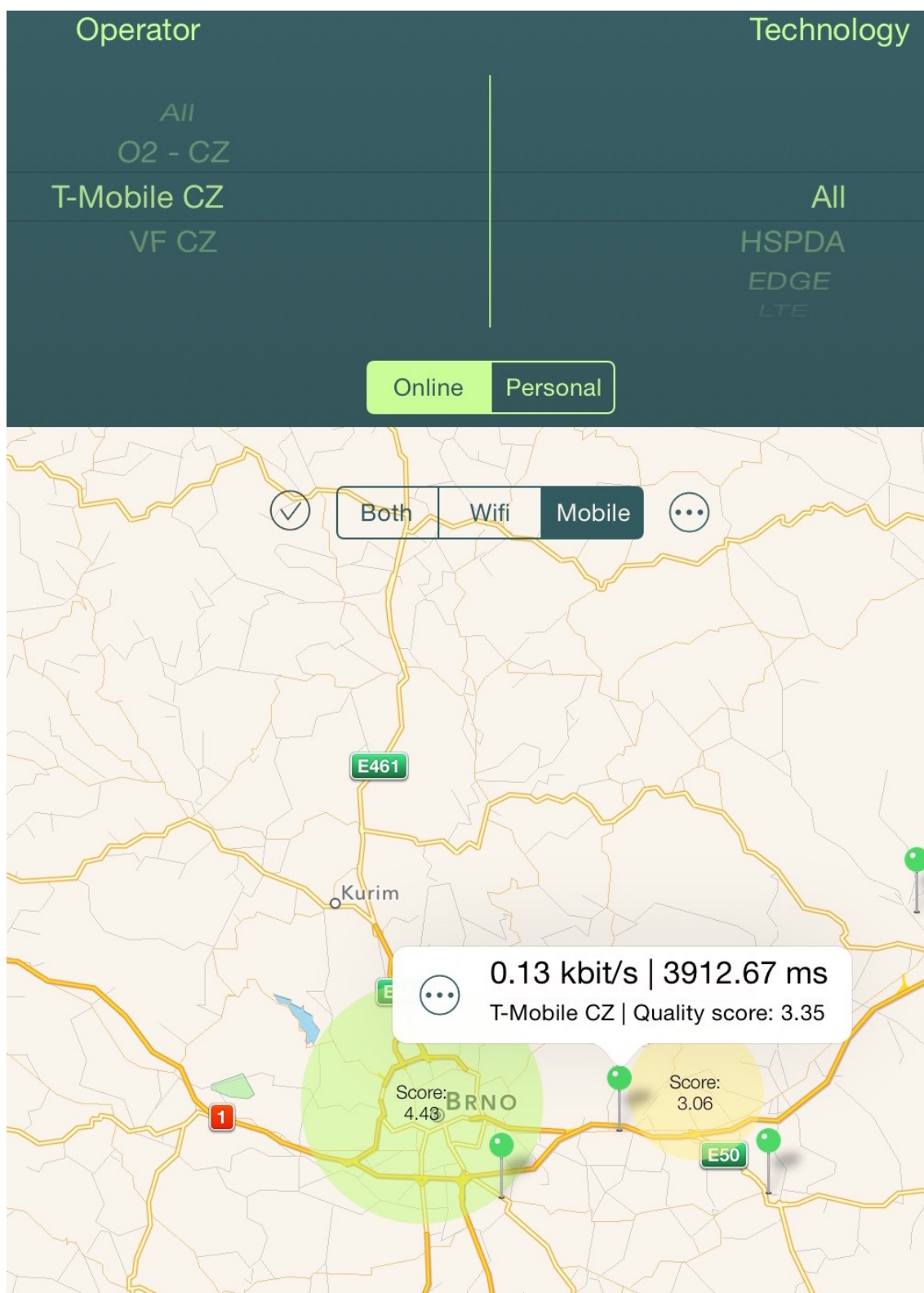
	WiFi	Mobilná sieť celkovo	Edge	HRPD	HSPDA	HSUPA	WCDMA	LTE
Maximálna rýchlosť [MBit/s]	94,17	61,71	0,216	0,495	28,82	6,444	23,854	61,71
Priemerná rýchlosť [MBit/s]	5,67	5,75	0,216	0,44	5,692	1,524	3,529	7,616

Počet stiahnutí aplikácie z App Store

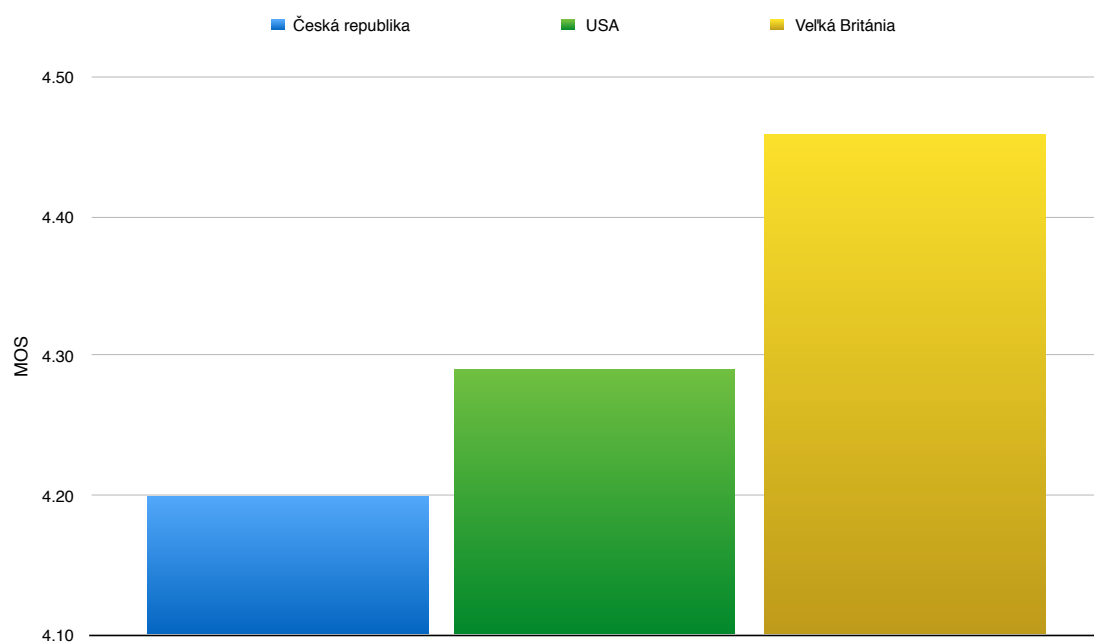
Celkovo	USA a Kanada	Afrika, Blízky Východ a India	Európa	Ázia	Latinská Amerika
4698	2470	1590	298	67	35



Obr. 5.5: Porovnanie českých operátorov v dosiahnutých výsledkov z meraní od užívateľov



Obr. 5.6: Porovnanie kvality internetu v meste a v jeho okolí



Obr. 5.7: Porovnanie priemernej kvality mobilného internetu v rôznych krajinách

6 ZÁVĚR

V rámci diplomovej práce bola vytvorená mobilná aplikácia pre platformu Apple iOS 7 a vyššie s podporou pre všetky modely iPhone a iPad. Táto mobilná aplikácia meria predpokladanú užívateľskú spokojnosť s kvalitou jeho pripojenia k internetu. Namierané hodnoty sú odosielané do online databáze z ktorej sa potom čerpajú dáta pri vytváraní mapy kvality pokrytia jednotlivými poskytovateľmi mobilného pripojenia. Táto mapa je tiež súčasťou tejto aplikácie.

Pri vývoji aplikácie som zároveň vytváral pomocné skripty pracujúce na strane serveru, ktoré komunikujú s databázou, kde sú uložené dáta z meraní a klientom mobilnou aplikáciou. Snažil som sa zabezpečiť správnosť údajov, ktoré server uloží do databáze a následne odosiela užívateľom do ich zariadenia. Išlo napríklad o roamingových užívateľov, ktorý svojimi meraniami pridávali body merania v zahraničí pod menom ich domáceho operátora. Preto server rieši dodatočne korektnosť prichádzajúcich dát od užívateľov.

Venoval som sa aj návrhu užívateľského rozhrania u ktorého som sa snažil, aby bolo čo najjednoduchšie a zrozumiteľné. Počas vývoja som riešil distribúciu aplikácie medzi testerov, pracoval som s vývojárskym portálom Apple, kde som musel vygenerovať certifikát pre distribúciu a podpísanie aplikácie z dôvodu jej distribúcie. Aby som získaval spätnú väzbu z aplikácií vo forme automatických reportov, implementoval som do aplikácie služby na to určené. Aj vďaka tomu som odhalil niekoľko problémových miest v aplikácii, kde dochádzalo k pádu aplikácie. Počas vývoja aplikácie boli zapojení do jej testovania niekoľko desiatok ľudí z celého sveta a aj na základe ich spätnej väzby som sa snažil aplikáciu pripraviť na distribúciu do App Store. Aplikáciu Apple do App Store schválil, a tak je volne k dispozícii pre všetkých užívateľov so zariadeniami iPhone alebo iPad pod názvom Internet Quality.

Z dát, ktoré namerali užívatelia som vytvoril grafy kvality internetového pripojenia pre rôzne druhy pripojenia a oblasti. Tiež som porovnal jednotlivých operátorov a vplyv dostupnosti LTE technológie medzi užívateľmi na celkové dosiahnuté hodnoty.

LITERATÚRA

- [1] HUIFANG, Chen; Yu Xin; Lei Xie, End-to-end quality adaptation scheme based on QoE prediction for video streaming service in LTE networks, "Modeling, Optimization in Mobile, Ad Hoc, Wireless Networks (WiOpt), 2013 11th International Symposium on , vol., no., pp.627,633, 13-17 May 2013 Dostupné z: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6576411&isnumber=6576384>
- [2] HUI, Zhang. ZTE. *QoE Modeling* [online]. 2012 [cit. 2014-10-26]. Dostupné z: http://wwen.zte.com.cn/endata/magazine/zte technologies/2012/no3/articles/201205/t20120522_307269.html
- [3] PENG YU, HUAIDONG ZENG, LANLAN RUI, WENJING LI a XUESONG QIU. A novel QoE assessment method for wireless networks. *2011 International Conference on Advanced Intelligence and Awareness Internet (AIAI 2011)* [online]. IET, 2011, č. 1, s. 323-327 [cit. 2014-10-26]. DOI: 10.1049/cp.2011.1482. Dostupné z: <http://digital-library.theiet.org/content/conferences/10.1049/cp.2011.1482>
- [4] MENKOVSKI, Vlado, Georgios EXARCHAKOS, Antonio LIOTTA a Antonio Cuadra SÁNCHEZ. Measuring Quality of Experience on a Commercial Mobile TV Platform. *2010 Second International Conferences on Advances in Multimedia* [online]. IEEE, 2010, č. 1, s. 33-38 [cit. 2014-10-19]. DOI: 10.1109/MMEDIA.2010.12. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5501601>
- [5] PERKIS, Andrew, Solveig MUNKEBY a Odd HILLESTAD. A model for measuring Quality of Experience. *Proceedings of the 7th Nordic Signal Processing Symposium - NORSIG 2006* [online]. IEEE, 2006, č. 1, s. 198-201 [cit. 2014-10-19]. DOI: 10.1109/NORSIG.2006.275209. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4052204>
- [6] KIST, Alexander A. a Lyn BRODIE. Quality of service, quality of experience and online learning. *2012 Frontiers in Education Conference Proceedings* [online]. IEEE, 2012, s. 1-6 [cit. 2014-10-19]. DOI: 10.1109/FIE.2012.6462223. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6462223>
- [7] ITU-T Recommendation P.800, "Methods for subjective determination of transmission quality, ITU, Tech. Rep., 1996

- [8] HOSEK, Jiri, Pavel VAJSAR, Lubos NAGY, Michal RIES, Olga GALININA, Sergey ANDREEV, Yevgeni KOUCHERYAVY, Zdenek SULC, Petr HAIS a Radek PENIZEK. Predicting user QoE satisfaction in current mobile networks. *2014 IEEE International Conference on Communications (ICC)* [online]. IEEE, 2014, č. 1, s. 1088-1093 [cit. 2015-05-10]. DOI: 10.1109/ICC.2014.6883466. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6883466>

7 OBSAH PŘILOŽENÉHO ARCHÍVU

V přílohe "QoE Network Test Analyzer.zip" sa nachádzajú všetky zdrojové kódy a návrh aplikácie. Spustenie projektu je možné pomocou programu XCode.